



Participation au Système d'Information de l'ISAE-Cnam Liban

Salim Rahal

► To cite this version:

Salim Rahal. Participation au Système d'Information de l'ISAE-Cnam Liban. Systèmes et contrôle [cs.SY]. 2011. dumas-01258769

HAL Id: dumas-01258769

<https://dumas.ccsd.cnrs.fr/dumas-01258769>

Submitted on 19 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE RÉGIONAL ASSOCIE DE BEYROUTH**

MEMOIRE

**présenté en vue d'obtenir
le DIPLOME D'INGENIEUR CNAM**

SPECIALITE : Informatique

OPTION : IRSM

par

Salim RAHAL

Tuteur

Mr. Pascal FARES

Participation au Système d'Information de l'ISAE-Cnam Liban

Soutenu le 12 Avril 2011

JURY

PRESIDENT: Prf. Jean-pierre ARNAUD

**MEMBRES: Mr. Elias EL HACHEM
Mr. Pascal FARES
Mr. Bachir HARB
Mlle. Nehmat TARTARIAN**

Remerciements

A la fin de cinq années d'études du diplôme d'ingénieur et deux années de travail à L'Institut des Sciences Appliquées et Économique (Centre du Liban associé au Conservatoire National des Arts et Métiers CNAM - Paris), voici le fruit de mon travail, accompagné de remerciements sincères et chaleureux.

Merci à Dieu qui m'a aidé !

Je remercie la direction de l'ISAE-Cnam Liban qui a été toujours là pour me donner l'appui et la force nécessaire pour accomplir ce projet.

Je remercie mon tuteur, Dr. Pascal Fares, chef du Département informatique et chef du projet de système d'information de l'ISAE-Cnam Liban, qui a accepté de diriger ce mémoire et qui m'a éclairé de ses précieux conseils.

De même, je remercie l'ingénieur Nehmat Tartarian, pour son assistance, et les autres collègues pour leur sympathie et la bonne ambiance qu'ils mettent.

Enfin, à tous ceux qui vont lire ce mémoire, j'espère qu'il en profitera. Merci à tous.

Liste des abréviations

ACID = Atomicity Consistency Isolation Durability
AJAX = Asynchronous JavaScript and Xml
AOO = Analyse Orienté Objet
ASP = Active Server Page
BCM = Business Cost Management
BIOS = Basic Input Output System
CGI = Common Gateway Interface
COO = Conception Orienté Objet
CRM = Customer Relationship Management
DCOM = Distributed component Object Model
DHTML = Dynamic HTML
Distro = Distrubutions
DOM = Document Object Model
DRM = Digital Rights Management
ECTS = European Credit Transfer
EJB = Enterprise Java Bean
ERP = Enterprise Resource Planning
ESB = Enterprise Server Bus
FOAD= Formations Ouvertes et à Distance
FSF = Free Software Foundation
FSI = Fournisseur de services Internet
FTP = File Transfer Protocol
GDN = Gestion Numérique des Droits
HTTP = Hyper Text Transfer Protocol
IDC = International Data Corporation
IIS = Internet Information Server
JDBC = Java Database Connectivity
JSF = java Server Face
JSP = Java Server Page
JVM = Java Virtual Machine
MDA = Model Driven Architecture
MVC = Model View Controller
NTIC = Nouvelle Technologies de l'Information et de la Communication
ODBC = Open Database Connectivity
OMT = Object Management Group
OMT = Object Modeling Technique
OOA = Object Oriented Analysis
OOD = Object Oriented Design
OSS = Open Source Software
PHP = Personal Home Page
PME = Petites et Moyennes Entreprises
PMI = Petites et Moyennes Industries
POC = Programmation Orientée Composant

POO = Programmation Orientée Objet
2TUP = Two Track Unified Process
RAD = Rapid Application Development
RMI = Remote Method Invocation
SGBDOO = Système de Gestion de Base de Données Orienté Objet
SOA = Service Oriented Architecture
UML = Unified Modeling Language
UP = Unified Process
URL = Uniform Resource Locator
USDP = Unified Software Development Process
VPN = Virtual Private Network
XP = Extreme programming

Table des matières

Introduction	10
I Environnement du travail	14
I.1 Structure de l'ISAE-Cnam Liban	14
I.1.1 Présentation de l'ISAE-Cnam Liban	14
I.1.2 Environnement organisationnel	15
I.1.3 Département SI	17
I.1.3.1 Emploi actuel	17
I.1.3.2 Activités et fonctionnalité du département	17
I.2 Ancien système	20
I.2.1 Anciens besoins	20
I.2.2 Ancien système	20
I.2.2.1 Principales caractéristiques du logiciel	21
I.3 La problématique	22
I.4 Nouveaux besoins	24
II Etat de l'art	26
II.1 Choix généraux	26
II.1.1 Pourquoi l'Open Source ?	26
II.1.1.1 Pourquoi choisir l'Open Source pour l'institut?	27
II.1.2 Pourquoi le Web?	28
II.1.2.1 Avantages de l'information sur le Web	28
II.1.2.2 Inconvénients de l'information sur le Web	29
II.1.2.3 Avantages et inconvénients d'un site web	30
II.2 Choix détaillés	30
II.2.1 Choix du système d'exploitation	30
II.2.1.1 Les avantages de Linux par rapport à Windows	31
II.2.1.2 Les inconvénients de Linux par rapport à Windows	35
II.2.1.3 Conclusion	35
II.2.1 Choix du langage de programmation	36
II.2.1.1 Apparition du langage Java	36
II.2.1.2 Les avantages de Java	36
II.2.1.3 Caractéristiques	38
II.2.2 Choix de la technologie Web	40

II.2.2.1 JSP.....	40
II.2.2.2 Autres langages dynamiques.....	41
II.2.2.3 Fonctionnement de JSP.....	42
II.2.2.4 JSP ou PHP ?.....	43
II.2.3 Choix de la base de données	44
II.2.3.1 Introduction.....	44
II.2.3.2 La présentation de JDBC (Java Database Connectivity)	44
II.2.3.3 Comparaison entre MySQL, PostgreSQL, Ingres, Firebird et Zodb	45
II.2.3.4 Justification du Choix MySQL à l'institut.....	48
III Environnement méthodologique.....	49
III.1 Conduite de projet	49
III.1.1 Besoin de la méthodologie	49
III.1.2 Évolution des approches de développement.....	49
III.1.3 Nécessité d'une méthode de développement.....	50
III.1.4 La méthode 2TUP.....	50
III.1.4.1 Définition d'un processus de développement logiciel.....	51
III.1.4.2 Le processus unifié	51
III.1.4.3 Le processus 2TUP adopté à l'institut	52
III.1.4.4 Un processus de modélisation avec UML	53
III.2 Programmation Orienté Objet (POO).....	54
III.2.1 Définition.....	54
III.2.2 Historique	55
III.2.3 Modélisation objet.....	55
III.3 Limitations de la programmation orienté objet	56
III.4 Programmation Orientée Composant (POC).....	57
III.4.1 Introduction	57
III.4.2 Définition.....	57
III.4.3 Structure d'un composant	57
III.4.4 Niveaux d'abstraction du POC	58
III.4.5 Les avantages à utiliser la POC	58
III.4.6 Conclusion	59
IV Analyse et conception.....	61
IV.1 Introduction.....	61
IV.2 Analyse.....	61

IV.2.1	Analyse de l'existant	62
IV.2.2	Conserver les bases du statut	62
IV.2.3	Intégrer des nouvelles notions	63
IV.2.4	Résoudre les anciennes fautes.....	64
IV.2.5	Problèmes rencontrés	65
IV. 3	Conception	65
IV.3.1	Architecture fonctionnelle du SI	66
IV.3 .1 .1	Diagramme de contexte (boîte noire)	66
IV. 3 .1 .2	Diagramme de contexte avec sous système fonctionnels (boîte blanche)	66
IV.3 .2	Les services à réaliser	67
IV. 3 .3	Graphe de flux de l'inscription	68
IV.3.4	Cas d'utilisation de l'inscription	69
V	Développement réalisé.....	71
V.1	Lancement du développement.....	71
V.2	Développement de la base de données.....	71
V.2.1	Préparations des serveurs SGBD	71
V.2.2	Création de la base de données	72
V.2.2.1	Modélisation.....	72
V.2.2.2	Moteurs de stockage.....	72
V.2.2.3	Créations des tables et des relations	73
V.2.3	Contrôle d'accès aux données	74
V.2.4	Bases de données distribuées	76
V.3	Développement Web	76
V.3.1	Introduction	76
V.3.2	Architectures.....	77
V.3.3	Les librairies.....	78
V.3.4	Utilisation d'AJAX	78
V.3.4.1	Définition	78
V.3.4.2	Implémentation au SI de l'ISAE-Cnam	80
V.3.5	Déploiement	81
V.3.5 .1	Choix du serveur d'application	81
V.3.5 .2	Importance du déploiement.....	81
V.4	Développement des composants	82
V.4.1	Du concept au développement	82

V.4.2	Avantage pratique	82
V.4.3	Architectures composants implémentées à l'institut	83
V.4.4	Source d'informations des composants	84
VI	Sécurité	86
VI.1	Introduction	86
VI.2	Sécurité des applications Web	86
VI.3	La sécurité des communications	87
VI.3.1	SSH	87
VII	Les fonctionnalités principales du produit	88
VII.1	Introduction	88
VII.2	Description globale	88
VII.2 .1	Les Modules	88
VII.2.2	Première version	88
VII.2.3	Deuxième version	89
VII.3	Description détaillée	90
VII.3.1	Module SI Étudiants	90
VII.3.1.1	Description du module « préinscription/inscription »	91
VII.3.1.2	Description du module « Horaire des Etudes »	93
VII.3.2	Module SI Enseignants	96
VII.3.3	Contribution personnelle	96
VIII	Développement des rapports	97
VIII.1	Recherche d'un outil des rapports	97
VIII.2	Architecture du « Jasper Report »	97
VIII.3	« IReport » (Designer)	97
VIII.4	Les fichiers	98
VIII.5	Intégration « Jasper » avec Web/MVC	99
IX	Bilan du projet	101
IX.1	État d'avancement	101
IX.2	Problèmes rencontrés	101
IX.2.1	Problèmes fonctionnels	101
IX.2.2	Problèmes techniques	102
IX.2.3	Temps de développement	102
IX.2.4	Amélioration possible	103
Conclusion		104

Bibliographie	106
Table des annexes	108
Annexe 1 Servlet – Application « sécurité sociale »	109
Annexe 2 Classe « UtilFunctions »	112
Annexe 3 Filter – Application « HoraireIsaeTC ».....	114
Liste des figures.....	117
Liste des tableaux	118

Introduction

Au niveau du monde entier, les applications Web touchent la plupart des secteurs financier, administratif, politique, militaire, technologique etc., c'est pourquoi, de nouvelles technologies apparaissent chaque mois et le besoin en informatique augmente de plus en plus au sein des entreprises et des institutions. En conséquence, «travailler dans le domaine NTIC » n'est pas assez facile, il faut être toujours au courant des divers évolutions.

Par ailleurs, l'Institut des Sciences Appliquées et Économiques (Centre du Liban associé au Conservatoire National des Arts et Métiers CNAM - Paris) a décidé de renouveler son système d'information et d'intégrer de nouvelles procédures au sein de son organisation en vue de satisfaire ses nouveaux besoins, surtout après son accroissement géographique au Liban et l'ouverture de différents centres d'enseignements (Baalback, Bekfaya, Ghazza, Tripoli, Baakline).

Afin de renouveler le système d'information de l'ISAE, deux propositions assez importantes sont proposées :

- Soit acheter un logiciel prêt à l'utiliser et par la suite, ce logiciel devrait être développé plus tard pour être adaptable aux spécifications spéciales de l'Institut.
- Soit développer notre propre système d'information dès le début.

Mais après l'estimation du coût du logiciel, de son développement et de son degré d'adaptation aux spécifications de l'ISAE, il était prévu donc de développer notre propre système d'information, et puisque nous possédons les qualifications nécessaires à développer ce système, et qu'on peut profiter des produits gratuits de l'*open source* (outil de développement, serveur Web, système opératoire, base de données etc.), afin de minimiser le coût de production d'un tel système.

En effet, plusieurs difficultés et problèmes ont été rencontrés au niveau de différents secteurs ; à titre d'exemple :

- L'analyse et la conception
Puisque l'ancien système présentait de divers failles à ce niveau, l'équipe Système d'Information (SI) a établi un nouveau cahier de charges pour ne pas se tromper de nouveau et conserver ses succès afin de faciliter les opérations et les procédures et non pas les compliquer.
- La technologie
Le choix du serveur d'application et de la base de données n'était pas assez facile car il existait au marché beaucoup de produits *open source* assez puissants. Il fallait donc choisir la meilleure solution et assurer l'interopérabilité entre les différents modules.
- L'administration
Certains centres d'enseignements étaient parfois complètement contre la centralisation des données par crainte de publier leurs données à l'extérieur du centre, pourtant ces centres n'avaient pas le droit de prendre une telle décision. Par

la suite, la direction de l'institut a imposé toutes ces décisions, ce qui a obligé alors ces centres à nous fournir toutes les informations nécessaires.

Toutes ces difficultés citées ci-dessus m'ont poussé, étant comme futur ingénieur, ainsi les autres ingénieurs, à trouver de nouvelles solutions. Par la suite, chaque solution a joué le rôle d'un motivateur qui nous a aidé à traverser toute difficulté. Par déduction, le travail sous pression et les difficultés rencontrées mènent tout ingénieur pour acquérir les compétences nécessaires en vue de devenir un vrai ingénieur !

Concernant ma contribution personnelle, j'ai participé à la programmation du SI de l'étudiant et de l'enseignant, d'où j'ai développé plusieurs parties :

- La gestion des horaires des professeurs.
- La fiche d'inscription qui comporte les tarifications des matières.
- La gestion des stages.
- La sécurité sociale.
- Le calcul des résultats des étudiants.

En ce qui concerne le module « catalogue des diplômes » j'ai réalisé l'analyse, la conception et la programmation de cette partie, ainsi que j'ai développé ce module de la couche base de données jusqu'à l'interface graphique.

Mon travail était sous la direction du chef de projet, du chef de développement, et avec la coordination de ma collègue développeuse.

Diverses tâches nécessitaient la coordination entre les membres de l'équipe, surtout celles qui touchaient le même module comme le cas de l'inscription, c'est pourquoi l'équipe utilisait le logiciel « CVS » qui gère le développement collectif. Par conséquent, après chaque développement de module j'effectuai une validation de mes codes, qui sera envoyée aux autres membres avec toutes les modifications et vise versa.

Enfin, autre que la conception et le développement, je suis arrivé à intégrer un nouveau logiciel «le IReport» pour développer des rapports de type PDF etc., et j'ai pu encapsuler les rapports avec nos applications Web pour être en ligne.

En effet, j'ai rédigé mon mémoire suivant trois pôles: gestionnaire, technique et méthodologique. Ces pôles expliquent les différents aspects de la réalisation de notre logiciel en vue de délivrer un produit maintenable et qui satisfait les besoins de l'institut. Par ailleurs, mon mémoire est constitué des parties principales suivantes:

- L'environnement du travail et l'état de l'art
- Le développement du SI de l'institut qui englobe les différentes étapes de développement du logiciel
- Le bilan du projet

Première partie

Le premier chapitre intitulé « Environnement du travail » est essentiel pour avoir des idées de la nature de l'institution et de ses services, ainsi que les différents départements, surtout celui du département SI dans lequel je travaille. Ce chapitre traite aussi l'ancien système afin de

dégager la problématique. Par la suite j'ai établi les nouveaux besoins qui vont imposer l'idée d'un nouveau SI.

Ensuite, le deuxième chapitre « État de l'art » va traiter la recherche d'une solution pour l'institut. En fait, avant de poser les solutions, j'ai étudié plusieurs choix possibles, qui peuvent être divisés en deux catégories :

- Les choix généraux concernent, d'une part, le choix d'utilisation de *l'open source* ou des logiciels propriétaires. D'autre part, le choix d'utilisation du Web à l'institut.
- Les choix détaillés concernent le type du système d'exploitation, Linux contre Windows, ainsi que le choix du langage de programmation, de la technologie Web et du SGBD.

Deuxième partie

Cette partie décrit la réalisation du projet, selon les chapitres suivants :

- III. Environnement méthodologique
- IV. Analyse et conception
- V. Développement réalisé
- VI. Sécurité
- VII. Fonctionnalité principale
- VIII. Développement des rapports

La base méthodologique est assez essentielle avant de commencer à réaliser un projet de développement d'un logiciel, dans ce cadre se situe le chapitre « III. Environnement méthodologique », ce chapitre traite surtout la méthode 2TUP (Two Track Unified Process) comme processus de développement et la Programmation Orientée Composant (POC) comme solution idéale au niveau modulaire, dans laquelle j'ai expliqué la structure d'un composant et les avantages de la POC.

Ensuite, j'ai réalisé une brève étude concernant l'analyse et la conception (chapitre IV) du SI de l'institut, j'ai surtout traité les problèmes rencontrés, l'architecture du nouveau système et j'ai donné l'exemple du cas d'utilisation de l'inscription.

Maintenant, c'est évident de parler du développement réalisé (chapitre V). Ce chapitre traite le développement des composants, en donnant des avantages pratiques de ce type de développement. J'ai parlé aussi du développement des applications Web MVC et l'intégration d'AJAX, en vue d'améliorer l'ergonomie des applications. Par suite, le chapitre intitulé « VI. Sécurité » traite la sécurité des applications à travers les filtres et la sécurité des communications via SSH.

Enfin, le fruit de ma participation est détaillée dans le chapitre intitulé « VII. Fonctionnalité principale » dans la section « VII.3.3 Contribution personnelle », où j'ai expliqué les différents services offerts par le logiciel. Par ailleurs, les rapports et leur intégration dans nos applications Web/MVC, ainsi que l'architecture intégrée sont expliqués au chapitre « VIII. Développement des rapports ».

Troisième partie

Le projet de renouvellement du SI de l'institut nécessite beaucoup de temps, d'où un bilan de ce projet est traité au chapitre intitulé «IX. Bilan du projet ». Dans ce chapitre, j'ai traité l'avancement du projet au cours des deux premières années et l'estimation afin de le terminer. De même, il traite aussi les erreurs fonctionnelles, techniques et propose des solutions possibles pour améliorer le SI suivant deux axes : fonctionnel et technique.

I Environnement du travail

I.1 Structure de l'ISAE-Cnam Liban

I.1.1 Présentation de l'ISAE-Cnam Liban

L'ISAE-Cnam Liban est un établissement d'enseignement supérieur proposant des formations continues diplômantes, ce qui permet aux salariés d'entreprises et d'industries d'évoluer dans leurs professions tout en gardant leurs emplois.

Depuis sa création en 1968, l'ISAE-Cnam Liban offre à ses auditeurs la possibilité de suivre des formations accréditées en France et reconnues au Liban. Ses objectifs se résument par :

- Développement de l'enseignement supérieur appliqué, dans des domaines techniques et professionnels divers, en collaboration avec des partenaires économiques et industriels libanais, pour assurer l'adéquation des formations avec l'évolution des métiers.
- Promotion de la formation continue afin d'améliorer les compétences des salariés dans les entreprises publiques et privées.
- Conception des formations centrées sur les compétences recherchées par les entreprises à travers des prestations sur mesure répondant aux besoins exprimés par ces entreprises.

La formation

Avant 2005 l'unité de valeur (UV) était adoptée, chaque unité d'enseignement correspond à 1 ou 1,5 UV, une UV correspond à 8, 10 ou 12 crédits.

Après 2005, la formation par crédit a été implémentée. Une unité d'enseignement correspond à 4 ou 6 crédits européens (ECTS – European Credit Transfer System). Un ECTS correspond en général à 10 heures d'enseignement. Une unité d'enseignement est dispensée pendant un semestre de 13 semaines à raison de 9 séances hebdomadaires, d'1h45 chacune, ou de 4 séances hebdomadaires de 1h30 chacune.

Objectif de la formation

L'étudiant peut s'inscrire à un nombre de crédits par semestre, actuellement 24 crédits dont 18 prioritaires au maximum et le reste comme choix. Si il dépasse les 18 crédits il peut y avoir des conflits dans les horaires de certaines matières et à l'examen aussi.

Le but d'inscription se fait soit pour obtenir un diplôme ou un certificat de compétence, soit dans le cadre d'une formation à la carte sans objectif du diplôme.

Dans tous les cas l'étudiant choisit son rythme de formation : la présence des valeurs théoriques est facultative (à l'exception des UE de niveaux Master), par contre celle des travaux pratiques est obligatoire.

Diplômes

Le Cnam propose des diplômes reconnus à tous les niveaux de l'enseignement supérieur, des diplômes de niveau Licence en sciences sociales, sciences économiques, sciences et techniques, des diplômes d'ingénieur, des diplômes de Master professionnel ou de recherche ainsi que des diplômes de doctorat. Au total, le Cnam délivre chaque année dans la France entière et à l'étranger plus de 7 000 diplômes, dont 1000 diplômes d'ingénieur.

La délivrance des diplômes Cnam est soumise à des exigences d'expérience professionnelle à partir des éléments transmis par l'auditeur dans le cadre de sa demande de diplôme. Une commission nationale émet un avis sur la nature, la durée et le niveau de cette expérience.

Actuellement, l'ISAE-Cnam Liban avec ces différents centres d'enseignement offre aux auditeurs des diplômes dans les spécialités suivantes :

- Informatique
- Économie et Gestion
- Électronique
- Électrotechnique
- Instrumentation : contrôle qualité
- Mécanique
- Énergétique
- Statistique
- Génie civil

Les diplômes offerts sont les suivants :

- Diplômes d'établissement : L'ISAE-Cnam Liban offre des diplômes de niveau Bac+2 et Bac+4 comme le RNCP (Répertoire National des Certificats Professionnels) de niveau II et III dans les différentes spécialités.
- Diplômes du système LMD : L'ISAE-Cnam Liban offre aussi des diplômes de niveau License en économie et gestion, statistique, informatique ; et des diplômes d'ingénieur en informatique, électronique, électrotechnique, mécanique et énergétique ; et aussi des diplômes Master en Gestion, contrôle et Audit, statistique, option : finance – assurance.

I.1.2 Environnement organisationnel

L'Institut est dirigé par le conseil d'administration dont le directeur est un membre. Le directeur exerce ses tâches administratives en coordonnant avec le comité de direction. Ensuite les conseils et les tâches sont issus aux différentes unités.

Les unités académiques englobent les départements et les cellules qui sont dirigés par des chefs. Chaque spécialité représente un département sauf l'électronique et l'électrotechnique qui sont fusionnés dans un seul département de même que la mécanique et l'électrotechnique. À

propos des cellules, il y a trois cellules à l'ISAE: Français, Anglais et Mathématique, qui sont dirigées par des chefs de cellules.

La figure suivante représente l'organigramme de l'ISAE-Cnam Liban:

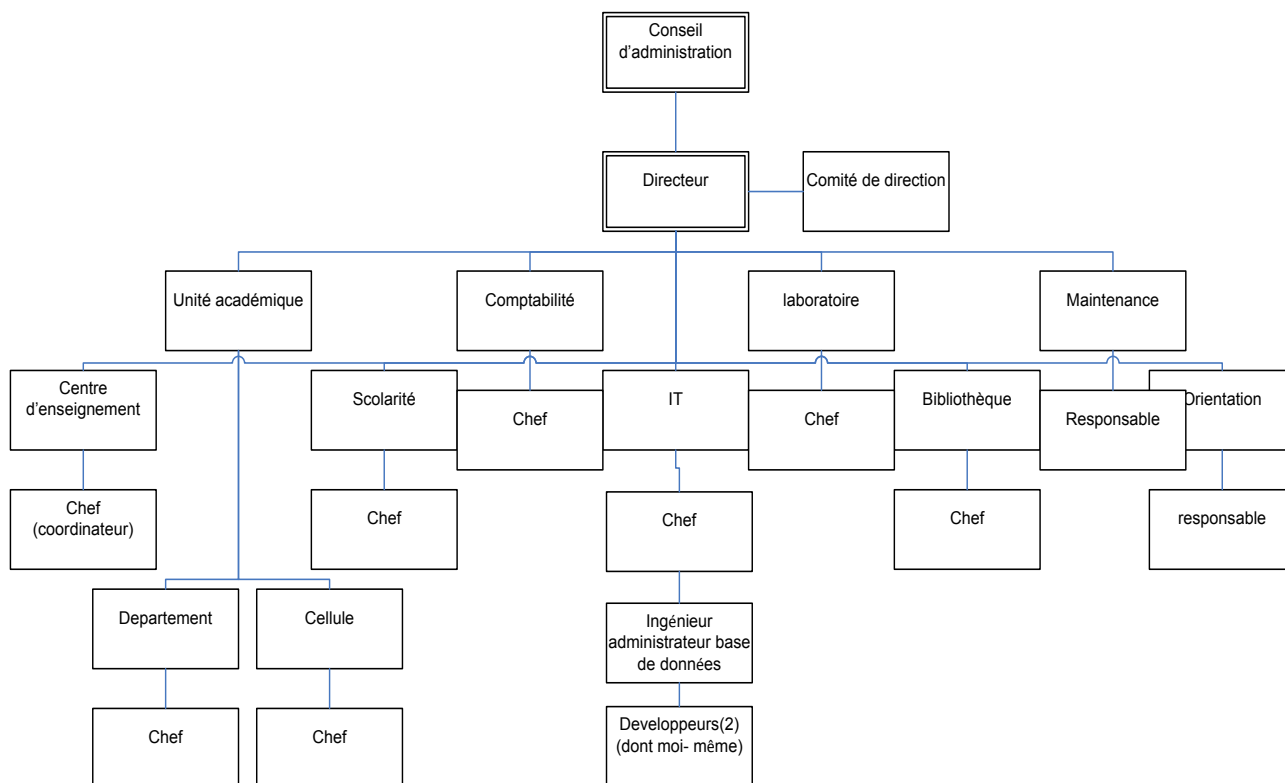


Figure 1: Organigramme de l'ISAE-Cnam Liban

Dans le cadre de s'étendre dans les différentes régions libanaises et avec la coopération des municipalités, l'ISAE-Cnam Liban a ouvert en 2008-2009 des centres d'enseignement dans les régions suivantes :

- Baakline (Mont- Liban sud), filières : Informatiques, Economie et gestion, Génie civil.
- Tyr(Liban), filière : Informatique, Economie et gestion.
- Baalback (Beqaa Nord), filières : Informatiques, Economie et gestion, Génie civil.
- Tripoli (Liban Nord), filières : Informatiques, Economie et gestion, Génie civil.
- Ghazza (Bekaa Sud), filières : Informatiques, Economie et gestion, Génie civil, Energétique.

- Bekfaya (Mont-Liban Nord), filières : Informatiques, Economie et gestion, Génie civil, Mécanique, contrôle qualité.

I.1.3 Département SI

Toute entreprise vise à développer ces procédures administratives et financières pour gagner du temps, améliorer la rentabilité de ses employés, contrôler et visualiser mieux les informations et satisfaire ses clients. Ce développement est l'implémentation de l'informatique et des nouvelles technologies dans tous les secteurs académiques, et financiers. Le SI est apparu en 1996, au début il été basé sur Microsoft Access et a été développé par un ingénieur, maintenant il est devenu en ligne (Web) et peut être accédé de n'importe quelle internaute connectée sur l'internet, bien sûre en respectant certains droits de sécurité. Avec l'évolution technologique du SI ou bien son renouvellement dont je voudrai traiter ultérieurement, le département SI s'est évolué aussi, maintenant il est administré par un chef, un ingénieur chef du développement et deux développeurs dont moi-même [figure 1].

I.1.3.1 Emploi actuel

Je travaille comme développeur de système à l'institut depuis 2/2/09.

Les tâches accomplies sont les suivantes :

- Analyse d'un système d'information proposé par le chef de projet ou le chef de développement.
- Conception, programmation et déploiement, tests de vérification avec la coordination avec les autres développeurs.
- Délivrance au chef de projet le logiciel réalisé.
- Maintenance du réseau Ethernet à l'institut.
- Réalisation des backups périodiques pour les bases de données.

I.1.3.2 Activités et fonctionnalité du département

Le département SI exerce diverses fonctions aux niveaux des différents secteurs. Ces fonctions sont :

Informatique

Le service technologie de l'information et de communication a une quadruple mission:

1. La gestion du parc informatique, des réseaux et de leurs sécurisations: les réseaux locaux dans chaque centre d'enseignement, l'interconnexion intranet entre les centres puis la gestion de l'accès à l'internet.

2. La gestion des plateformes, des ressources et de l'architecture de la présence en ligne (web) de l'ISAE Cnam Liban :
 1. Les applications en collaboration avec Google Apps (Email, Calendrier, Documents en ligne et partagés sur le nuage Google, les outils collaboratifs groupes et sites).
 2. Les applications open source Apache, TikiWiki (wiki, articles, blog et forum), PhpBB (forum), Moodle (plateforme FOAD) et Tomcat.
3. La spécification et le développement en interne des applications, de leurs plateformes de développement et de déploiement formant le système d'information de l'ISAE-Cnam Liban: Gestion ressources humaine, gestion des auditeurs, gestions des enseignements, gestion des résultats, gestion des diplômes, gestion de la planification et des horaires, gestion des examens.
4. Étude et implémentation d'une solution d'enseignement à distance, en temps réel ou différé, utilisant au maximum l'infrastructure intranet et internet.

Administratif et communication :

Le département SI communique avec tous les autres départements et exécute le plan proposé par l'administration, qui est représentée par le directeur de l'institut et le chef du projet. Les départements avec lesquels le département SI communique sont les suivantes :

- Comptabilité
- Scolarité
- Unité académique
- Direction
- Centre d'enseignement
- Liban Post

La partie suivante vous donnent une description de la relation du département SI avec les autres départements avec lesquels il communique :

Le service comptabilité consulte le département SI et vice-versa, pour obtenir diverses informations à propos des salariés et des professeurs et leurs contrats etc. Aussi, des applications spécifiques sont mises en œuvre pour la comptabilité et la direction, comme le SI des enseignants qui traite les contrats des professeurs. D'autre part, le service scolarité qui assure des services pour les étudiants et gère leurs dossiers, il est en contact permanent avec l'équipe du SI surtout dans la période d'inscription et d'examen ; par exemple le transfert d'un dossier d'étudiant d'un centre à un autre se fait par le département SI après la demande de la scolarité.

En suite, des applications sont misent en places pour les unités académiques, pour gérer leurs spécialités, leurs diplômes et leurs catalogues sur le Web. Les chefs des unités académiques ou de départements, sont dotés de plusieurs tâches sur le SI, surtout durant la période d'inscription des étudiants. Les essentielles tâches d'un chef de département sont : l'accord à l'inscription à certaines matières, la gestion des horaires, affectation des professeurs etc.

Concernant la direction, qui est représentée par le directeur, le vice directeur et le secrétaire administratif. Elle exerce son travail, comme la consultation, la supervision et la proposition des projets à réaliser. Par ailleurs, la direction consulte le département SI pour des statistiques, par exemple : avoir des rapports sur les étudiants inscrits dans tous les centres. De même, la direction supervise notre travail de deux manières : directe et indirecte. La manière directe consiste à visualiser les opérations sur le Web, grâce à des applications spécifiques, l'indirecte est basée sur le « feedback » des employés et des étudiants qui contactent la direction pour un raison après avoir utiliser le système en Web, par exemple : un enseignant qui demande une révision sur ses heures enseignées. Par suite, le meilleur service offert par le SI, est le contrôle simultané des différentes opérations, surtout ce qui concerne le budget et les salaires.

Concernant les centres d'enseignements, il sont représentés par des chefs et des secrétaires administratives. Les centres sont connectés à l'administration centrale grâce au SI. Des divers services sont offertes pour eux, par exemple, des statistiques à propos des étudiants inscrits et préinscrits pour un semestre donné ; fixer des erreurs d'inscriptions sur la fiche de paiement (annuler certaines matières, éditer l'inscription). Techniquement l'intranet qui nous relie avec les centres d'enseignement doit être dans des bonnes conditions, elle est gérée par l'entreprise « Cedercom », qui fournit des services réseaux et infrastructure. Donc le département SI joue un rôle central sur plusieurs niveaux administrative et technique.

Le Liban Post est une entreprise pour les services postaux, cette entreprise est dotée de recevoir le cash des étudiants, comme frais d'inscription et frais de sécurité sociale puis transférer les montants payés vers le compte bancaire de l'institut. Ensuite le Liban post fournit des rapports de format électronique (emails) à l'institut concernant les étudiants qui ont payé leurs frais. Notre équipe informatique valide l'inscription des étudiants en se basant sur ces rapports. Par suite, après la validation des inscriptions, les étudiants peuvent demander des attestations et passer leurs examens normalement. Mais des problèmes sont rencontrés quand il y a un retard dans la délivrance des rapports de Liban Post concernant les étudiants inscrits, surtout quand ils demandent des attestations etc. Enfin, comment on peut les servir si nous ne connaissons pas s'ils ont payés leurs frais d'inscription ? La solution est de demander du Liban Post de nous fournir des rapports journaliers pour que notre SI soit cohérent le maximum avec le système du Liban Post, ce qui se passe actuellement.

Activités internes et formations :

Des réunions périodiques sont élaborées pour l'équipe informatique. Le chef du projet détermine le sujet de la réunion qui sera discuté, et à la fin nous arrivons à une conclusion qui peut être décisive, par exemple après discussion nous arrivons à adapter l'architecture orientée service(SOA) dont je voudrais expliquer ultérieurement. Aussi, Des formations étaient organisées par l'Agence Universitaire de la Francophonie (AUF) à Tripoli, ces formations sont assistées par des cadres de différentes universités et compagnies, et auxquelles nous avons participé. Bien sûr ce type de formations a amélioré notre compétence technique et notre niveau de relations humaines.

I.2 Ancien système

I.2.1 Anciens besoins

Avant 2008, le centre de Beyrouth était le seul au Liban. Ses employés comptaient à peu près 40 employés, nombreux facteurs ont influencés l'implémentation de l'ancien système qui est apparu en 1996, nous citons à titre d'exemple :

- La zone géographique limitée
- Le volume des données limitées
- Le débit internet limité
- Le besoin d'un SI simple
- L'utilisation du système opératoire Windows au sein de l'institut

Toutes ces caractéristiques ci-dessus sont prises en compte (de point de vue matériel, logiciel et fonctionnel) dans la réalisation de l'ancien système, autrement dit, nous avons installé un système adaptable sur LAN, car notre zone de travail était bornée, et ce système n'utilisait pas l'internet car le débit était faible. Et puisque tous les utilisateurs utilisaient le Windows donc nous n'avons pas besoins d'un logiciel portable. Enfin, fonctionnellement nous avons besoin d'un système simple non « complexe » capable de gérer l'institut comme un seul centre d'enseignement, donc pas de données distribuées mais centralisées.

D'autre part les besoins sont variés et les processus métiers aussi, or nous pouvons les classer comme suivants :

- **Gestion des étudiants :**

Qui englobe les services offerts aux étudiants comme l'inscription, période de l'inscription, les résultats, les examens, les sessions, les horaires, les catalogues des diplômes, admission.

- **Gestion des enseignants :**

Contrats des enseignants, présence, surveillance, correction, salaires des enseignants.

- **Gestion du stock :**

Gestion des matériels consommables et non consommables.

- **Gestion des ressources humaines :**

Information des employés, salaires, congés, présence, maladies.

- **Gestion de la petite caisse**

Gestion des factures et des cash échangés.

I.2.2 Ancien système

En basant sur les besoins cités dans le paragraphe précédant, le système Access a été adapté depuis 1996 comme logiciel de gestion de l'ISAE-Cnam Liban.

Microsoft Access ou MS Access (officiellement Microsoft Office Access) est un logiciel comprenant un système de gestion de base de données relationnelle édité par Microsoft. Il fait partie de la suite bureautique MS Office Pro.

MS Access est composé de deux programmes : le moteur de base de données et l'éditeur graphique. Les deux ne sont pas librement séparables.

I.2.2.1 Principales caractéristiques du logiciel

MS Access est un logiciel utilisant des fichiers au format Access (extension de fichier mdb pour Microsoft DataBase). Il est compatible avec les requêtes SQL et dispose d'une interface graphique pour saisir les requêtes. Il permet aussi de configurer, avec des assistants ou librement, des formulaires Query et sous-formulaires de saisie, des états imprimables (avec regroupements de données selon divers critères et des totalisations, sous-totalisations, conditionnelles ou non), des pages html liées aux données d'une base, des macros et des modules VBA. [16]

Comme beaucoup de systèmes de gestion de bases de données relationnelles, ces données peuvent être utilisées dans des programmes écrits dans divers langages.

Les langages couramment utilisés avec Access sont le Visual Basic for Application (VBA) et les langages qui disposent de modules d'accès aux données pour les fichiers .mdb : Delphi de Borland, Visual Basic, C++ sous Visual Studio de Microsoft par exemple. VBA, intégré à Access comme à toutes les applications de la suite Microsoft Office, permet de créer des applications de gestion complètes, livrées avec un programme d'installation qui gère automatiquement la mise en place éventuelle d'un « runtime » d'Access, et dont le code source est protégé dans une version semi-exécutable des fichiers (mde).

Il est en effet possible (et conseillé) d'installer un fichier mdb, contenant les tables de données, sur un serveur (ou un poste de réseau poste à poste dédié comme tel) et des fichiers mdb (ou mde) contenant tous les éléments de l'application sur les postes client. Cette séparation permet le déploiement de mises à jour de la partie applicative sans avoir à transvaser les données existantes d'une base à une autre. Dans ce cas, les fichiers clients sont « attachés » aux tables du fichier « mdb » installé sur le poste serveur. Sur ce plan, Access est donc très simple à déployer car il n'a pas besoin de service serveur comme MySQL, Oracle, PostgreSQL, etc. D'un côté comme de l'autre, ce fichier étant un conteneur, il faudra s'assurer de sa viabilité. En effet, au moindre défaut du fichier (défaut d'écriture disque, etc.) les chances de perte de l'application ou des données sont très grandes et la réparation automatique du fichier conteneur tiendra presque du miracle. Un modèle de sauvegarde est donc fortement recommandé quelle que soit l'utilisation d'une base Access.

D'après Microsoft, MS Access supporterait des configurations de 256 postes. Mais en pratique, pour une utilisation confortable, MS Access serait limité à une vingtaine d'utilisateurs simultanés, les échanges réseaux étant 10 à 20 fois plus gourmands en ressources qu'avec Microsoft SQL Server par exemple. En effet chaque application Access aura accès au fichier de données à travers le réseau sans passer par une application serveur qui limite le flux de données au strict nécessaire suivant la requête exécutée. Le confort d'utilisation de MS Access en réseau peut-être considérablement accru quand on utilise judicieusement les requêtes de type Snapshot (lecture seule, modifications ultérieures non visibles) et les requêtes de type Dynaset (lecture-écriture) qui sont beaucoup plus gourmandes en ressources réseau. Ces perfectionnements ne permettent pas de dépasser pratiquement une quarantaine d'utilisateurs en simultané. Il est important de noter que ces caractéristiques conviennent largement à la plupart des petites et moyennes entreprises.

D'un point de vue concret, Access (avec ses versions 2000 à 2003) convient bien à des applications faisant intervenir jusqu'à une centaine de tables (principales et de jointures / relations) avec un maximum pratique de 100 000 enregistrements pour les tables principales et de 1 000 000 d'enregistrements pour les tables de jointures (appelées aussi tables de liaisons ou de relations).

Pour une utilisation de plus grande envergure, il peut servir de client pour un serveur de bases de données (comme SQL Server, Oracle, MySQL, etc.) via ODBC ou OLE DB, on parle d'utilisation frontale. Si Access est limité en nombre d'utilisateurs, il peut par contre gérer, sur de petits réseaux locaux de bonne qualité technique, des quantités d'informations qui vont bien au-delà des besoins de beaucoup d'organismes. Exemples : une base de données des pièces automobiles de toutes les marques pour des garages, ou une comptabilité sur un seul site, mais autorisant des consolidations de plusieurs sites autonomes.

Les données d'Access sont facilement exploitables dans les publipostages de Word et les tableaux Excel. Réciproquement les feuilles de données d'Excel peuvent être « attachées », comme une des tables de la base de données ou importées ponctuellement dans une table Access.

Par le biais du langage de programmation VBA, il est possible, depuis l'intérieur d'Access, de construire ou de modifier tout type de tableaux Excel, puis de traiter et transférer les informations adéquat de la base de données vers ce même tableur. Attention une fois Excel lancé depuis Access, il faut le fermer proprement. Sinon l'application peut continuer à tourner en tâche de fond, sans être visible de l'utilisateur.

En Java, Microsoft Access peut être utilisé de façon transparente via JDBC à l'aide de la passerelle JDBC-ODBC de Sun.

I.3 La problématique

L'Access avait servi l'institut à peu près 11 ans depuis 1996, il avait présenté plusieurs avantages comme la facilité de programmation surtout qu'il est adaptable aux applications simples pour les entreprises à tailles petites et il comporte un outil des rapports (Built-in). Mais L'Access présentait aussi plusieurs failles même avant l'accroissement de l'ISAE-Cnam au Liban et l'ouverture des centres, nous citons à titre d'exemple :

- La taille de la base de données étant limitée par 2 GB.
- Les échanges réseaux étant gourmandes en ressources.
- L'incompatibilité entre certaines versions comme 2003 et 2007, pas de mise à jour du logiciel.
- L'Access est non portable, fonctionne seul sur Windows.
- La vulnérabilité du Windows aux Virus et attaques, ce qui rend nos postes en danger.
- L'absence d'architecture (n-tiers..) des programmes, ce qui les rend difficiles à maintenir, car la maintenance des composants est plus facile que la maintenance d'un produit monolithique.
- Le coût de License de l'Access, par contre il y a des logiciels gratuits au marché.

En effet, toutes ces limitations n'ont pas arrêté le développement et le fonctionnement du SI à l'institut, mais le problème essentiel était la non adaptation du SI aux nouveaux besoins surtout lors de l'ouverture des six centres d'enseignements. Puisque l'Access est le logiciel de

gestion de l'institut, donc il est installé dans chacun des centres. Il y avait des postes serveurs et clients pour chaque centre d'enseignement, donc sept serveurs à base de données et des dizaines des postes clients distribués sur les différents centres, notons que chaque centre travaillait indépendamment de l'autre car l'Access ne fonctionnait pas entre des différents LAN, due à ses limitations (maximum 20 postes) comme j'ai déjà mentionné au début du paragraphe, et n'oublions pas que le débit de l'intranet est limité entre les centres. Pour ces raisons le développement des applications et la mise à jour de bases des données étaient des opérations sophistiquées pour les raisons suivantes :

- La mise à jour de bases de données des centres était une opération manuelle et périodique, en leur envoyant des CDs qui contiennent les mises à jour. Cette méthode posait beaucoup de problèmes, comme la duplication de bases de données et leurs incohérences dues au délai de la mise à jour entre Beyrouth et les centres et réciproquement [figure2].
- La modification au niveau des applications doit être envoyée aux centres, donc on a aussi la duplication des applications.
- Des modifications peuvent être effectuées par les centres et qui ne sont pas contrôlés par le centre principal de Beyrouth, soit au niveau de bases de données soit au niveau des applications.

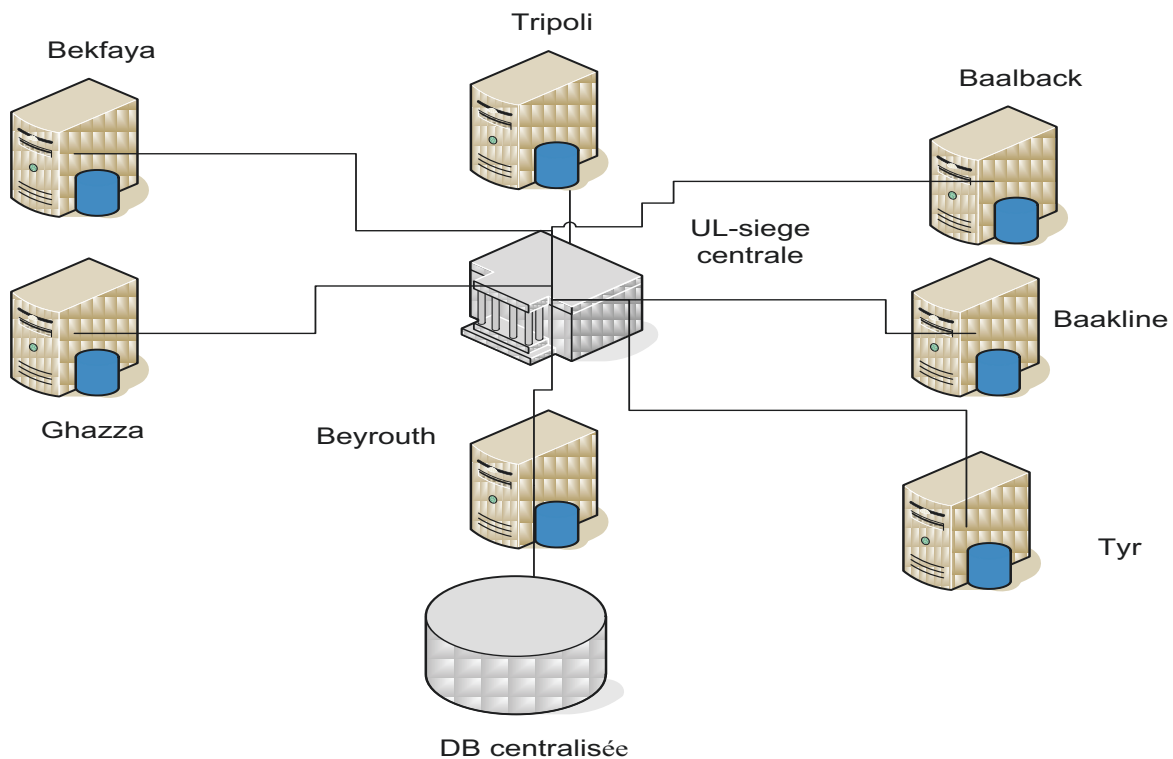


Figure 2: Duplication des bases de données aux centres

Faces aux problèmes des duplications des données et des applications et autres problèmes déjà cités, nous allons étudier les nouveaux besoins dans le chapitre suivant.

I.4 Nouveaux besoins

Nous avons déjà expliqué dans le chapitre précédent les anciens besoins qui ont abouti à l'ancien système « l'Access », mais plusieurs facteurs ont déclenchés la recherche d'un nouveau SI, citons les suivants :

- L'ouverture des nouveaux centres d'enseignement a disposé de nouveaux besoins administratifs et technologiques.
- Le problème d'incohérence dans certains modules. Par exemple, dans le SI des étudiants et des enseignants, ce qui expliquait l'importance d'étudier les besoins de nouveau dans le but d'implémenter des nouveaux modules.
- La non satisfaction de certains besoins essentiels, par exemple la procédure de l'inscription.
- La duplication de la Base de données. Des copies sont distribuées pour tous les centres.
- La duplication des applications.
- La sécurité est insuffisante.
- Le non flexibilité du SI existant avec les nouveaux besoins.

Après avoir présenter les facteurs qui ont imposé l'étude d'un nouveau SI, donc nous pouvons résumer les besoins suivants :

- Un système de gestion de base de données qui doit permettre à l'équipe d'effectuer efficacement des contrôles périodiques, y compris la centralisation de la base de données, la traçabilité et le suivi. Aussi ce système envisagé doit respecter certaines normes comme la rapidité des accès : il doit pouvoir fournir les réponses aux requêtes le plus rapidement possible, cela implique des algorithmes de recherches rapides.
- Des applications centralisées et modulaires, développées suivant une méthodologie, afin de faciliter les opérations de la maintenance et l'évolution des programmes.
- Un système portable, c'est-à-dire opérable sur tous les systèmes d'exploitation (Windows et linux..).
- Une politique de sécurité bien définie utilisée aux différents secteurs suivants:
 - Réseaux : par exemple, sécuriser les connexions en utilisant le protocole SSH.
 - Applications : protéger nos applications Web en appliquant plusieurs procédures(chapitre VI- Sécurité).
 - Bases de données : le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs.
- Un système cohérent, pas de conflit entre les modules, cela nécessite une nouvelle étude du système d'information avec une nouvelle vision basée sur les nouveaux critères et ressources.
- Nouveau système d'information de l'étudiant : l'inscription des étudiants doit être par diplôme, il faut informatiser les catalogues des diplômes et organiser les matières suivants les diplômes et les spécialités, dans l'ancien SI les étudiants ne s'inscrivaient pas

dans des diplômes, seulement dans les matières, le nouveau SI prévu doit changer ce concept tel que les étudiants connaîtrons dans quel diplôme ils sont inscrits.

- Facilité de manipulation : on cherchait aussi une facilité dans la manipulation du système. Par exemple, la procédure de création des utilisateurs à l'Access était gênante car pour chaque application on doit créer un utilisateur. Il était évident qu'on devait trouver une solution à toute cette procédure.
- Facilité d'utilisation un nouvel interface graphique : une meilleure interactivité avec le système était préférable pour mettre l'utilisateur à l'aise en lui assurant un meilleur confort dans son travail surtout avec l'utilisation d'un interface graphique beaucoup plus développé que celui de l'Access par exemple. Ainsi, un employé serein donnerait beaucoup plus, surtout avec un esprit tranquille.
- Système en ligne : nous avons besoin d'un système qui travaille 24h/24, accessible aux employés et étudiants, en assurant les ressources nécessaires pour le fonctionnement permanent des équipements. Par exemple, résoudre le problème du découpage de l'électricité.

Dans le chapitre II - Etat de l'art, nous allons traiter les choix techniques adoptés dans notre projet avec les justifications de choisir tel et tel produit du marché.

II Etat de l'art

II.1 Choix généraux

II.1.1 Pourquoi l'Open Source ?

Définition [1]

La désignation *open source* (code source libre) s'applique aux logiciels dont la licence respecte des critères précisément établis par l'*open source* Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et de travaux dérivés.

Souvent, un logiciel libre est qualifié d'*open source*, car les licences compatibles *open source* englobent les licences libres selon la définition de la FSF.

Le terme *open source* est en concurrence avec le terme « *free software* » recommandé par la FSF. Le terme « *freeware* » (gratuitiel) désigne des logiciels gratuits qui ne sont ni nécessairement ouverts, ni libres.

Marché de l'*open source* [2]

Les logiciels *open source* s'installent dans les mœurs. Selon IDC (International Data Corporation pour les analyses intelligentes du marché d'IT), le secteur a représenté 1,8 milliard de dollars en 2006. Cette embellie devrait se confirmer dans les prochaines années, avec un chiffre d'affaires estimé à 5,8 milliards de dollars pour 2011, ce qui représente une croissance annuelle d'environ 26%.

Les logiciels libres se mettent au payant

Cette croissance devrait être portée par l'engouement des entreprises pour des logiciels libres commerciaux, moins chers et plus souples à l'utilisation que leurs concurrents propriétaires. Malgré ces résultats prometteurs, le secteur des OSS est encore loin d'être mûr. Si les systèmes propriétaires - hors services à la demande - sont vendus à la licence, les logiciels libres sont encore souvent proposés selon un système d'abonnement progressif, ou ne sont facturés que sur les seuls services de support. Difficile dans ce contexte de générer des revenus importants...

Pour Matt Lawton, directeur de programme pour l'Open Software Business Models d'IDC, le secteur est encore dans ses "premières phases de développement".

Les acteurs du marché

Le marché mondial du logiciel *open source* est dominé par trois acteurs ; le géant IBM - qui a contribué pour plus d'un milliard de dollars au développement et à la promotion du système d'exploitation Linux -, Red Hat et Novell, les deux éditeurs des distributions Linux (Red Hat Linux et Novell Suse Linux) les plus présents sur le marché.

Pour autant, ils sont très loin d'être les seuls, et on assiste à l'arrivée de grands acteurs comme Oracle et SAP.

Les segments de l'*open source* professionnelle :

Il y a cinq segments dans lesquels les entreprises investissent dans l'open source :

- Les outils de développement, développement d'applications, collaboration, gestion du cycle de vie des projets ;
- Les serveurs web et d'applications, plates-formes et serveurs applicatifs, middleware, ESB, suites d'intégration, produits de messagerie instantanée, portails ;
- Les bases de données;
- Les applications d'entreprise, bureautique, ERP (gestion), CRM (clients), email, BCM (gestion de contenu), BI (Business Intelligence) ;
- Les systèmes d'exploitation.

L'exemple d'IBM est significatif, le géant de l'informatique est présent sur tous ces segments, à la fois avec des solutions propriétaires et avec des solutions open source.

Voici une liste non exhaustive des logiciels libres les plus utilisés par application et leurs équivalents propriétaires :

Application	Logiciels libres	Logiciels propriétaires
Navigation internet	Mozilla Firefox	Internet Explorer
Système d'exploitation	GNU/Linux	Microsoft Windows
Bureautique	Open Office	Microsoft Office
Messagerie	Thunderbird	Outlook, Lotus
Messagerie instantanée	aMSN, Pidgin	Windows Live Messenger
Graphisme	GIMP, Sodipodi	Photoshop, Illustrator
Editeur HTML	Nvu	Dreamweaver
Création site	SPIP, Joomla	Documentum
Création blog	Wordpress	Typepad
Antivirus	Avast	Norton
Antispam	SpamBayes	VadeRetro
Comptabilité/Gestion	PHPCompta	Ciel Compta

Table 1: Listes des logiciels libres et leurs équivalents propriétaires

II.1.1.1 Pourquoi choisir l'Open Source pour l'institut?

Pour plusieurs raisons, et voilà dix points :

1. Parce que c'est gratuit
2. Parce que ça marche et que c'est performant
3. Parce que nous pouvons l'adapter à nos besoins
4. Parce qu'il n'y a pas de spyware, ni de pourriture

5. Parce qu'il y a une communauté derrière qui connaît le produit et qui peut vous dépanner mieux que les entreprises qui ne vous donneront jamais les défauts de leurs produits
6. Parce que les failles de sécurité sont détectées plus vite
7. Parce qu'il y a souvent des mises à jour dans les grands projets
8. Parce que les concepteurs n'ont pas d'argent à gagner en vendant des fausses mises à jour
9. Parce que c'est facile à prendre en main sans aucune formation puisque ça ressemble aux logiciels courants.

Pour toutes ces raisons, l'institut a décidé d'adopter l'open source en général - avant d'entrer dans les détails - comme infrastructure logicielle.

II.1.2 Pourquoi le Web?

Après avoir traité les nouveaux besoins de l'institut dans le chapitre « Environnement de travail », l'ouverture des nouveaux centres d'enseignement et le besoin de l'accessibilité de l'information sont des facteurs majeurs qui ont imposés l'adoption d'une nouvelle architecture logicielle, d'où vient le Web, était la solution possible. Étudions dans le paragraphe suivant les avantages et les inconvénients de l'information sur le web pour l'institut.

II.1.2.1 Avantages de l'information sur le Web

- Accès à l'information
 - Accès à distance à l'information, d'université, d'un endroit public, du domicile de l'étudiant ou de celui d'amis et coéquipiers.
 - Accès par plusieurs internautes en même temps à une page Web (très utile pour un travail en équipe).
- Disponibilité de l'information : 24 heures par jour, 7 jours par semaine.
- Recherche
 - Recherche thématique (répertoires) et par mots-clés (moteurs de recherche).
 - Avec les moteurs, recherche de plusieurs mots-clés en une seule opération.
 - Accès rapide aux termes recherchés dans le texte intégral des pages Web repêchées (CTRL + F).
- Contenu
 - Information souvent impossible à trouver sur les autres supports.
 - Mise à jour rapide par l'auteur ou le « WebMaster ».

- Notes de lecture : Possibilité de copier-coller des paragraphes pour ajouter aux notes de lecture, d'enregistrer une page Web ou de l'imprimer en tout ou en partie.
- Fichiers multimédias
- Navigation en hypertexte : certains liens renvoient à un approfondissement du sujet.
- Large choix d'images gratuites (cartes, schémas, photographies, cliparts, etc.) pour enregistrer, modifier, importer dans un travail, imprimer.
- Sons et séquences vidéos (parfois gratuites) pouvant être incorporés dans une présentation.
- Communication. Le courriel permet de correspondre avec les coéquipiers, une équipe d'une autre école, le prof, des experts externes. Ouverture sur le monde sans contrainte géographique.
- Discussion, partage d'information et entraide sur divers aspects d'un projet dans un forum de discussion ou dans un blogue, créé par le prof ou par une équipe.

II.1.2.2 Inconvénients de l'information sur le Web

- Recherche
 - Équations de recherche. L'utilisateur doit savoir comment chercher l'information, et doit noter ses équations pour ne pas recommencer les mêmes recherches.
 - Contexte d'utilisation. La plupart des moteurs repêchent des résultats sans distinguer le sens des termes recherchés. Ex. « chat » trouve chat (animal) et chat (service de messagerie). Il faut parfois ouvrir plusieurs pages Web pour trouver le sens recherché.
- Contenu
 - Le contenu n'est pas toujours fiable et il est même parfois médiocre
 - Beaucoup de pages Web ne sont pas à jour
 - Pages supprimées, adresses modifiées sans préavis : certains liens sont rapidement désuets
- Structure : pas de structure logique comme dans un livre, mais le Plan du site peut aider à comprendre la structure.
- Liens hypertexte reliant le texte d'une page Web au texte d'autres pages Web : il est facile de perdre la trace de départ.
- Texte déroulant qui fait perdre de vue les écrans et les textes antérieurs. L'utilisateur n'a pas de vue d'ensemble d'un sujet, comme dans un livre, mais plutôt plusieurs écrans.
- Publicité envahissante sur certains sites, nuisant parfois à la concentration.
- Problèmes techniques qui, parfois, ne peuvent être réglés que par un technicien en informatique.

- Coût : Il faut être abonné à un FSI à la maison, au travail.

II.1.2.3 Avantages et inconvénients d'un site web

- Avantages :
 - Un site web pour PME et PMI peut permettre l'ouverture de marchés hors d'une zone imitée par la géographie.
 - Il permet également de développer un contenu dynamique et de créer une véritable interaction entre les partenaires d'une société.
- Inconvénients :
 - Un site web a un coût en terme :
D'hébergement
De création
De référencement
De communication

Ensuite, il faut bien distinguer deux types de sites :

- Le corporatif, simple vitrine commerciale qui assure une présence sur internet et donne la possibilité de créer une carte de visite avancée. Toutefois pour certaines activités, ce genre de site est totalement dénué d'intérêt.
- Le e-business est une alternative très intéressante dans le cadre du développement d'une PME. Elle coûte nettement plus chère à la création et à la maintenance qu'une corporatif, mais permet une ouverture totale de l'activité sur des audiences nationales ou internationales.

En fin de compte, l'ISAE- Cnam Liban a choisi la solution Web dès 2008 pour satisfaire ses besoins, surtout qui sont apparus avec l'ouverture des plusieurs centres d'enseignements et avec l'augmentation des volumes des informations comme nous avons expliqué dans l'environnement de travail, d'où l'ancien système MS ACCESS ne les supportait pas. En effet, des facteurs majeurs ont disposés le choix du Web, comme l'accès à distance à l'information et sa disponibilité sur le web sur n'importe quel poste, soit au Liban soit à l'étranger.

II.2 Choix détaillés

II.2.1 Choix du système d'exploitation

Une fois la décision de l'utilisation du Web et de produits Open Source a été prise, il faut maintenant choisir le système d'exploitation convenable, je voudrai traiter le système opéra-

toire Linux avec ses avantages et ses inconvénients par rapports au système fermé Windows et dans la conclusion de ce paragraphe je voudrai souligner notre choix à l'ISAE-Cnam Liban.

II.2.1.1 Les avantages de Linux par rapport à Windows

Voilà les avantages essentielles de linux par rapport à Windows [3] :

1- Aucun virus et « spyware » (fichiers espions) n'affectent les fichiers ou programmes systèmes de manières critiques ou dangereuses. Dans Linux, on n'a pas besoin d'anti-virus ni d'anti-spyware sauf si c'est pour scanner des fichiers sur une partition dans un réseau partagé avec d'autres systèmes d'exploitations Windows. Si les virus et les « spywares » sont si efficaces contre Windows, c'est que ce dernier contient d'énormes trous de sécurité et de faiblesses dans le système qui n'ont jamais été corrigés par Microsoft. Et les virus ou plutôt les « crakers » informatiques (ceux qui conçoivent les virus et spyware) savent comment exploiter ces failles du système. Il existe quelques virus contre Linux (on peut les compter sur les doigts de la main) mais ceux-ci ne sont pas considérés comme dangereux.

2- Existe plusieurs sortes de distributions Linux (plus de 200). Alors, là, il y en a pour tout les goûts. Il y a des distro Linux orientés spécifiquement pour le "multimédia", d'autres orientés "serveurs ou sécurités", orientés "développeurs ou programmeurs", orientés "bureautique", orientés "pour tester", orientés "pour les jeux", orientés pour "l'éducation (établissement scolaire)", orientés "pour scientifiques", etc. En somme, on choisit la distribution qui nous convient le mieux. Beaucoup de distro Linux peuvent englober plusieurs "spécialités" décrites ci-haut. Chaque distro a son propre concept de gestion des systèmes, de logiciels ou/et de manières d'utilisations. Les distros les plus populaires à l'heure actuelle sont: Ubuntu, Fedora Core, Mandriva, Debian, Suse, Knoppix et Gentoo. Ils ont chacun leur propre public cible: par exemple, Ubuntu s'adresse à un large public ne connaissant pas ou peu le système Linux tandis que la distribution Gentoo s'adresse à des utilisateurs expérimentés. Chacune des distributions a sa propres communauté de développeurs (concepteurs). Par conséquent, chaque distro a ses propres avantages et inconvénients. Remarque: Linux n'est pas un système lié à une distribution spécifique. Si un problème survient dans une distro, l'erreur en revient à la distro concernée et non à Linux en général.

3- Evolution rapide des distributions les plus connues (en moyenne, une nouvelle version majeure tout les six-sept mois pour les distro les plus populaires !).

4- En général, pas besoin d'installer des pilotes matériels (sont inclus dans le noyau Linux, le cœur du système) mais cela dépend de la version du noyau et de l'âge du matériel utilisé. Dans la plupart des cas, la détection matériels se fait de manière quasi-parfaite. Ne pas oublier que plus on utilise du matériels récents, plus on devrait utiliser un noyau récent afin d'augmenter les chances de détections et de configurations stables. L'inverse est également vrai pour de vieux ordi, on doit alors utiliser autant que possible un vieux noyau. Mais remarquer que cela n'est pas une règle absolue. À ce propos, nous avons une petite observation à vous faire: Le noyau Linux a été conçu d'une manière dite de retro-ingénierie ou dite en anglais "reverse

engineering". La plupart des fabricants de matériels informatiques ne rendent pas public le code source de leurs programmes pilotes (drivers). Alors, si on tient du fait que les développeurs du noyau Linux n'ont pas accès aux codes sources de ces fichiers pilotes, vu le succès actuel du noyau Linux dans la détection matériels, il faut considérer ces développeurs comme étant de vrais génies informatiques. Les efforts encourus ici sont très considérables et très exigeants. Très souvent, le noyau Linux exploite certaines fonctionnalités peu exploitées voire cachées de la machine mais n'en sont pas moins importantes, ce qui va sans dire qu'il faut un sacré génie pour découvrir ces fonctions (peu exploitées même par les fichiers pilotes propriétaires). Pour ma part, nous avons souvent remarqué que certaines périphériques (prévues à la base pour Windows) fonctionnent mieux sous un noyau Linux que sous Windows (même si dans certains cas on doit faire quelques manipulations dans certains fichiers système). Cela a pour effet aussi d'améliorer la tolérance aux pannes du système.

5- Il y a plus de supports techniques et de documentations gratuites sous Linux que dans Windows et ce, dans toutes les langues connues. Les réponses données dans les forums Linux en général sont techniquement "plus souples", "plus pointues et détaillées" dans le cas des problèmes sérieux, les solutions aux problèmes ne se limitent pas aux seuls "clic par çi", "clic par là", etc. Observation dans les forums pour Windows: En général, les réponses données dans ces forums sont "moins techniques" ou "techniquement moins professionnels" que les forums Linux. La plupart des réponses sont du genre "clic sur ce bouton", "ouvre tel fenêtre", etc, ce qui en soit n'en apprend rien sur le vrai fonctionnement du système car tout dans Windows est fermé.

6- Plus de choix de logiciels libres et gratuits sous Linux que sous Windows. Le choix des logiciels sous Linux est très impressionnant. La meilleure preuve est de vérifier le nombre total de logiciels libres pour Windows et Linux sur un serveur de programmes comme par exemple Sourceforge.net, l'un des plus gros serveur mondial de logiciels multi-plateformes. Plusieurs de ces logiciels deviennent vite à l'état de maturité (stabilité, performance, sécurité et professionnalisme) comme par exemple le logiciel GIMP, OpenOffice, Blender, Firefox, Apache, Iptables, Webmin, XMMS, Mplayer, Amarok, GPG, Inkscape, Rezound, etc.

7- Dans Linux, il n'y a pas de signature numérique DRM (en anglais *Digital Rights Management*) ou gestion numérique des droits ou GDN en français. La signature numérique DRM consiste à contrôler la lecture d'une oeuvre numérique. Elle déterminera si vous avez ou non le droit de voir ou de modifier l'œuvre (un fichier texte, vidéo, image, logiciel, ...etc). À noter que la version de Windows, Windows 7, intègre la signature numérique DRM et ce, pour tout ce qui sera dans le système. Pour de plus amples informations sur le DRM (en français), voyez le site Wikipédia.

8- Les extensions des fichiers ne sont pas obligatoires (comme .txt, .bin, .doc,...) pour être fonctionnelles dans le système. Néanmoins, avec les environnements graphiques comme KDE ou Gnome certains types de fichiers ont besoin d'une extension pour être utilisés graphiquement.

9- Est beaucoup moins gourmand en ressources systèmes (RAM, CPU, espace disque,...) que Windows, ce qui a aussi comme conséquence une plus grande rapidité d'exécutions des pro-

grammes que ce dernier. Une gestion de la mémoire paginée (la « swap ») est mieux exploitée afin que la mémoire vive ne soit pas toujours saturée. La swap (ou partition d'échange) dans Linux est située sur une partition disque indépendante de la partition système. Dans Windows, la swap est contenue dans un fichier d'échange installé dans la même partition que Windows.

10- A) Accepte beaucoup plus de requêtes sollicitées à la seconde (serveurs réseaux ou lancement de plusieurs applications locaux en même temps) sans planter ou/et sans redémarrage du système.

Un exemple courant: le serveur web Apache qui est beaucoup moins gourmand en ressources système que son concurrent IIS du côté Windows. La raison est que Apache n'utilise pas d'interface graphique. À noter que le logiciel Apache existe aussi pour Windows.

B) Linux est un vrai système multitâches: on peut lancer plusieurs applications sans que le système ne ralentisse ou que la mémoire ne devienne saturée. Par exemple, vous pouvez ouvrir un fichier vidéo avec tel logiciel et ouvrir le même fichier vidéo en même temps avec un autre logiciel sans qu'il y ait blocage de la lecture du fichier. Windows est aussi un système multitâches mais la gestion des processus et de la mémoire utilisée par les applications sont beaucoup moins efficaces que Linux. Qui n'a pas entendu parler d'un Windows qui "gèle" ? On ne peut pas en dire autant du côté Linux.

11- Gestions de plusieurs types de partitions Linux et Windows (fat16, fat32, ntfs, ext2fs, ext3fs,, xfs,...). Cependant, même si Linux est capable d'écrire et de gérer les données sur une partition NTFS (selon la version du noyau utilisé), il n'est pas recommandé de le faire. NTFS est une partition native à Windows (XP, 2000 server, 2003 server) et non à Linux. De ce fait, il est donc plus pratique de laisser gérer une partition NTFS par Windows plutôt que par Linux.

12- Pas besoin de défragmenter les partitions de type Linux (ext2fs, ext3fs, reiserfs, xfs,...). En fait, Linux ne fragmente pas les fichiers, ce qui avantage énormément les vitesses d'accès et d'écritures aux fichiers. De plus, un disque dur sous Linux aura une durée de vie plus longue que sous Windows car le disque a beaucoup moins tendance à "tourner en fou" pour accéder ou écrire des données.

13- Est parfaitement fonctionnel quand Linux est installé en mode terminal (c'est-à-dire sans interface graphique). En fait, l'interface graphique est optionnelle. Même si la plupart des utilisateurs utilisent (ou veulent) une interface graphique comme KDE ou Gnome, il est tout de même bon de ne pas oublier que Linux, par principe, n'en a nullement besoin pour bien fonctionner. Comme un rappel, au tout début de l'histoire de Linux, il n'y avait pas d'interface graphique à l'écran.

14- L'interpréteur de commande (le terminal) n'est pas intégré de manière fixe au noyau ce qui donne l'avantage de pouvoir faire sa mise-à-jour sans changer tout le système. Linux est multi-blocs (les environnements graphiques, le terminal, les services et les programmes installés sont indépendants les uns des autres) tandis que Windows est monobloc (l'interface graphique, le terminal MS-DOS ainsi que les services du système sont dépendants les uns des autres).

15- Les commandes dans un terminal peuvent interagir avec les programmes graphiques comme par exemple ouvrir OpenOffice mais à partir du terminal. De ce fait, s'il doit y avoir un

problème avec un logiciel graphique quelconque lors de son ouverture , le résultat des erreurs (débogage) apparaîtra dans le terminal (s'il y a lieu) ce qui aidera l'utilisateur à mieux déceler et à réparer la nature du problème. Très pratique pour un bêta-testeur de logiciels graphiques en version test.

16- Les bibliothèques graphiques des programmes à interfaces graphiques, ont été développées indépendamment du programme principale attaché. Exemple : Le gestionnaire de paquetage Synaptic fait appel aux commandes apt-get et à dpkg. L'avantage de cette manière de programmer avec une application graphique, est que le développeur se concentre d'avantage sur son programme de base (sans interface graphique) alors qu'un autre développeur ne s'attarde que sur l'aspect graphique (bibliothèque(s)) de l'application rattaché. Attention, ce ne sont pas toutes les applications graphiques qui sont développées de cette façon. Il y a aussi beaucoup d'applications graphiques dans Linux qui ne peuvent fonctionner sans bibliothèques graphiques (comme par exemple Firefox, OpenOffice, etc). Un autre avantage, plus important, est que si une application graphique "plante" ou "gèle", on peut toujours continuer à utiliser le programme en mode console (à condition que le bug soit celui de la bibliothèque graphique, ce qui est souvent le cas).

17- Un crash d'un environnement graphique (comme KDE ou Gnome) n'entraîne pas un crash du système. L'interface graphique est totalement indépendante du système.

18- Possibilité de télécharger et de gérer des milliers de logiciels via les outils de mise à jour du système (qui est personnalisable) comme l'utilitaire Synaptic pour les paquetages Debian (.deb).

L'installation des paquetages dans Linux est plus uniformisée (voire organisée) que celle dans Windows (.exe, .msi), ce qui contribue aussi à la stabilité de Linux et des logiciels installés. Explication : Dans Windows, les logiciels ont chacun leurs méthodes d'installations de leurs fichiers, ce qui les rend moins "uniformisées" les uns des autres par rapport à Windows. Plusieurs de ces installations ont tendance à créer certains conflits de fichiers (ou même de "thread") dans le système ou envers certains programmes, ce qui a pour conséquence que l'on doit réinstaller certains logiciels (à court ou à long terme).

19- Une sécurité plus sévères et efficaces sur l'accessibilité des programmes (droits d'accès). À ce propos, ne jamais se logger par défaut en mode root (comme c'est le cas par défaut avec Windows XP) car si vous avez accès à tout les programmes, d'autres personnes malintentionnées soit sur le net ou en local peuvent le faire. Ce qui n'est pas très sécuritaire. Cela dépend bien sûr de la configuration de la "sécurité" installée. Du côté des Windows, la version offrant la meilleure sécurité système à ce jour est *Windows server 2003*.

20- A une plus grande tolérance aux pannes que Windows (surtout pour les versions récentes du noyau). Les "plantage" de Linux sont beaucoup moins fréquentes que Windows et ce, peut importe la version du noyau utilisé. C'est ce qui constitue principalement le grand succès de Linux auprès des entreprises dans le monde.

21- Gestion et administration de tout le système à distance via l'utilisation d'un navigateur internet (en passant par exemple par l'utilitaire graphique webmin) ou par un terminal avec la commande SSH.[4]

II.2.1.2 Les inconvénients de Linux par rapport à Windows

1- Linux est plus complexe que Windows. En effet, Linux possédant une structure plus complexe demande de la part du Linuxiens débutant une plus grande recherche et de bidouillages dans ce dernier pour s'y retrouver et ce, même s'il y a beaucoup de documentations conçues pour lui. Cela demande bien sûr, de nouvelles habitudes d'utilisation et aussi beaucoup de temps d'apprentissage. Cela dit, ce n'est pas un gros inconvénient car il y a toujours des solutions à tout les problèmes : voyez les forums, blogs, site web spécialisés consacrés au débutants Linuxiens. De plus, la difficulté d'utilisation de Linux dépend fortement de la distribution utilisée. Certaines distro sont plus faciles à apprendre que d'autres: comme par exemple Ubuntu, Mandriva, Xandros,...[4]

2- Le nombre impressionnant de distributions Linux peut dérouter le débutant Linuxien. Comme il y a plus de 200 distributions Linux dans le monde, (et il s'en ajoute à tout les mois), il y a vraiment de quoi s'y perdre. On ne sait pas toujours quel distro convient le mieux pour tel ou tel besoin.

3- Pas assez de publicité pour mieux faire connaître Linux dans le monde. À part Internet, il n'y a que très peu de publicité faites par des entreprises, mêmes par ceux utilisant régulièrement Linux. Y a-t-il beaucoup d'annonces dans les journaux sur Linux ?

4- Un manque de pilotes ou de drivers propriétaires pour Linux. Le fait que la plupart des constructeurs de matériels informatiques ne puissent écrire des fichiers pilotes pour Linux n'aident pas beaucoup aux concepteurs de noyau à mieux détecter et exploiter au maximum toutes les fonctions de certains périphériques comme les imprimantes, scanners, routeurs Wi-Fi, webcam, etc.

5- Les gros jeux commerciaux manquent cruellement à Linux. En effet, Windows est mieux connus pour les jeux commerciaux de tout genre. Plusieurs utilisateurs avancés de Linux disent qu'ils préfèrent Windows pour les jeux et Linux pour le reste...Et pourtant, lorsque l'on regarde en profondeur tout le potentiel que peut offrir Linux sur les jeux (son noyau, OpenGL, ainsi que Wine, Cedega, Xwine, ...), Linux offre autant de capacité sinon plus que Windows pour faire rouler de gros jeux commerciales (à condition qu'ils soient conçus pour Linux).

II.2.1.3 Conclusion

Par déduction, on peut arriver à constater que Linux offre plus de potentiels à exploiter que Windows. Concernant les inconvénients de Linux mentionnés ci-haut, on peut constater aussi que tout cela ne sera que temporaire, car tout dans Linux évolue vite. Beaucoup plus vite que Windows. Et les défauts d'hier ne seront pas les mêmes défauts de demain (même s'il y aura de nouveaux problèmes). On n'a qu'à se rappeler les premières distributions au tout début de l'histoire de Linux: très compliquée et instable, beaucoup de bidouillages à faire, installation difficile, documentation manquante pour la plupart des applications, nombreux problèmes de détections matérielles, pas très intuitif, les logiciels à interfaces graphiques mature sont rares,

etc. Mais tout ça a énormément changé aujourd'hui, et pour le meilleur. Comme quoi, Linux complète à merveille ce qui manque à Windows, et vice-versa.

En conclusion, je dis qu'il n'y a pas de OS parfait mais plutôt des OS plus parfait que d'autres. Tout dépend de nos besoins. Pour toutes ses raisons le département SI a adopté l'Ubuntu comme système opératoire du serveur web et de la base de données, et moi comme développeur j'ai adopté Ubuntu dès mon début professionnel à l'institut.

II.2.1 Choix du langage de programmation

Le langage le plus connu au niveau des logiciels Open Source est Java, nous allons justifier le choix du Java comme le langage de programmation au sein de l'institut dans cette partie, en soulignant ses avantages et ses caractéristiques. Java n'est pas seulement un langage, c'est une plate-forme logicielle qui est conçue surtout pour les applications internet et contrairement aux autres langage comme le C/C++. Des concurrents existent au marché comme la langage C#, mais nous avons éliminé ce langage aussi puissant de nos discussions car notre choix est déterminé dans le chapitre II.1 « Choix Généraux », c-à-dire d'utiliser les technologies de l'Open Source.

JAVA

PLATE-FORME

II.2.1.1 Apparition du langage Java

Vers la fin de 1995, le langage de programmation Java surgit sur la grande scène d'Internet et obtient immédiatement un énorme succès. La prétention de Java est de constituer la colle universelle capable de connecter les utilisateurs aux informations, que celles-ci proviennent de serveurs Web de bases de données, de fournisseurs d'informations ou de toute autre source imaginable. Et Java se trouve en bonne position pour accomplir ce pari. Il s'agit d'un langage de conception très performant qui a été adopté par la majorité des fournisseurs. Ses caractéristiques intégrées de sécurité offrent un sentiment de confiance aux programmeurs comme aux utilisateurs des applications. De plus, Java incorpore des fonctionnalités qui facilitent grandement certaines tâches de programmation avancées comme la gestion des réseaux, la connectivité des bases de données ou le développement d'applications multitâches. La réputation de Java en tant que langage informatique est exagérée: Java est assurément un bon langage de programmation et aussi une bonne matière d'enseignement comme au sein de l'ISAE-Cnam Liban d'où Java constitue la base technologique essentiel de nos étudiants.[5]

II.2.1.2 Les avantages de Java

L'un des avantages évidents de ce langage est une bibliothèque d'exécution qui se veut indépendante de la plateforme: en théorie, il vous est possible d'utiliser le même code pour Windows 95/98/NT, Solaris UNIX Macintosh, etc. Cette propriété est indispensable pour une pro-

grammation sur Internet (cependant, par rapport à la disponibilité sur Windows et Solaris les implémentations sur d'autres plates-formes ont toujours un léger décalage).

1) Architecture classique avec un bytecode différent pour chaque processeur (figure03).

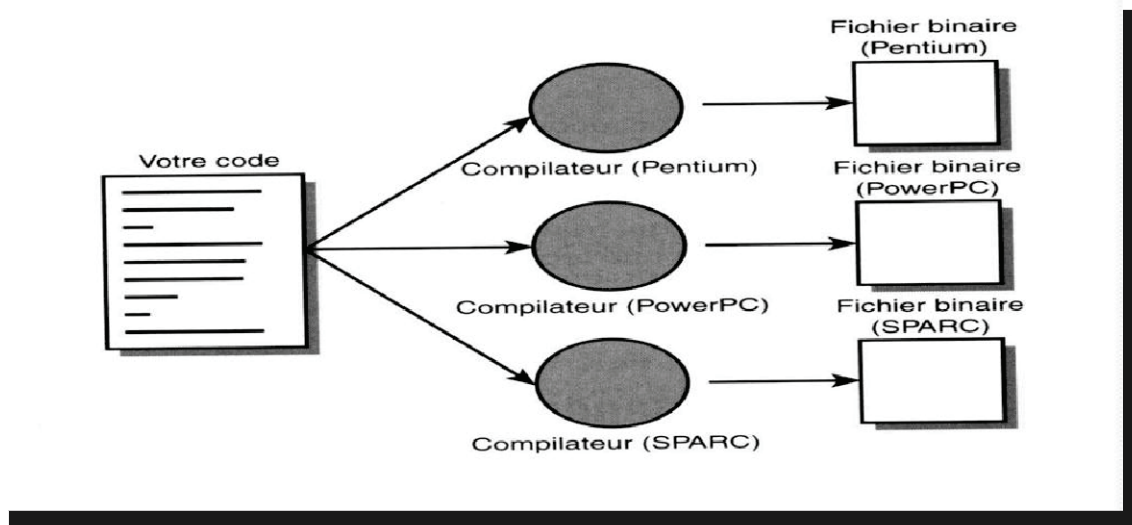


Figure 3: Architecture classique

2) Architecture Java, le bytecode passe par l'intermédiaire d'un interpréteur(figure04).

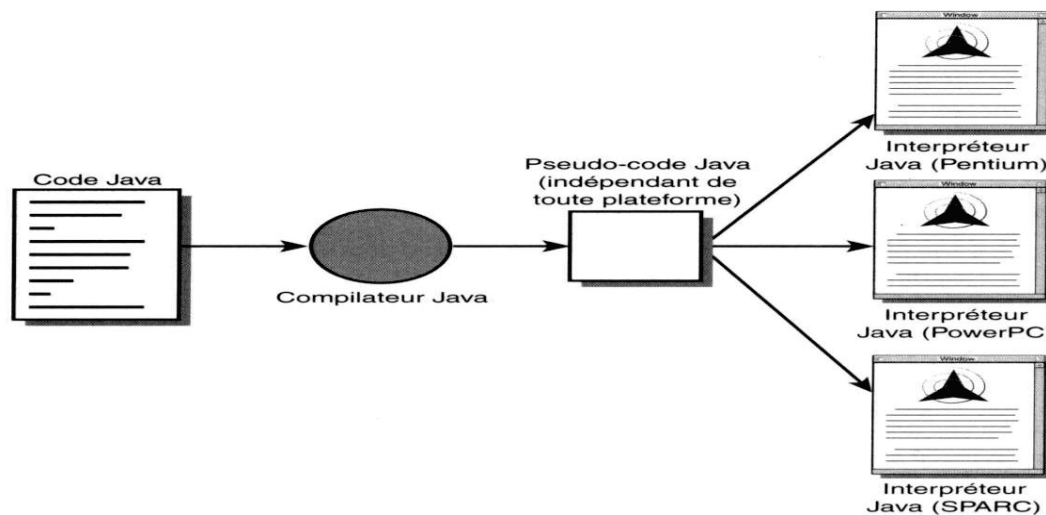


Figure 4: Architecture Java

Un autre avantage de ce langage de programmation réside dans le fait que la syntaxe de Java est analogue à celle de C++ ce qui le rend économique et professionnel.

Le fait de créer une autre version d'un langage C++ n'est cependant pas suffisant. Le point clé est le suivant : il est beaucoup plus facile d'obtenir du code sans erreur à l'aide de java qu'avec C++

Pourquoi ? Les concepteurs de Java ont beaucoup réfléchi à la raison pour laquelle le code C++ contenait autant d'erreurs. Cette réflexion les a amenés à ajouter dans Java des fonctions destinées à éliminer la possibilité de créer du code contenant les types d'erreurs les plus courants (selon certaines estimations, le code C++ contient au moins une erreur toutes les cinquante lignes).

- Les concepteurs de java ont supprimé l'allocation et la libération de mémoire manuelles. La mémoire dans java est allouée et libérée automatiquement. Vous n'avez jamais à vous préoccuper de pertes de mémoire.

- Ils ont éliminé l'arithmétique des pointeurs introduisant du même coup une vraie gestion de tableau. La notion de référence sur une zone mémoire remplace avantageusement celle de "pointeur", car elle supprime la possibilité d'écraser toute zone mémoire à cause d'un compteur erroné.

- Ils ont éliminé toute possibilité de confusion entre une affectation et un test d'égalité dans une instruction conditionnelle. Une instruction if (entries - 3) ne pourra pas franchir l'étape de la compilation

- Ils ont supprimé l'héritage multiple en le remplaçant par une nouvelle notion d'interface dérivée d'Objective C. Les interfaces vous offrent tout ce que vous pouvez obtenir à partir de l'héritage multiple, sans la complexité de la gestion de hiérarchie d'héritage multiple.

II.2.1.3 Caractéristiques

Les créateurs de Java ont écrit un livre blanc qui présente les caractéristiques fondamentales de Java. Ce livre est articulé autour des 11 termes suivants [5]:

- Distribué

Java possède une importante bibliothèque de routines permettant de gérer les protocoles TCP/IP tels que HTTP et FTP. Les applications Java peuvent charger et accéder à des fichiers sur Internet via des URL avec la même facilité qu'elles accèdent à un fichier local sur le système.

« Nous avons trouvé que les fonctionnalités réseau de Java sont à la fois fiables et d'utilisation aisée. Toute personne ayant essayé de faire de la programmation pour Internet avec un autre langage se réjouira de la simplicité de Java lorsqu'il s'agit de mettre en œuvre des tâches lourdes, comme l'ouverture d'une connexion avec un socket. De plus, Java rend plus facile l'élaboration des scripts CGI et un mécanisme élégant, nommé « Servlet », augmente considérablement l'efficacité du traitement côté serveur, assuré par Java. De nombreux serveurs Web, parmi les plus courants, supportent les « Servlets ». Le mécanisme d'invocation de méthode à distance (RMI) autorise la communication entre objets distribués. »

- Fiabilité

Java a été conçu pour que les programmes qui l'utilisent soient fiables sous différents aspects. Sa conception encourage le programmeur à traquer préventivement les éventuels problèmes, à lancer des vérifications dynamiques en cours d'exécution et à éliminer les situations génératrices d'erreurs... La seule et unique grosse différence entre C++ et Java réside dans le fait que ce dernier intègre un modèle de pointeur qui écarte les risques d'écrasement de la mémoire et d'endommagement des données.

- Orienté objet

Pour rester simples, disons que la conception orientée objet est une technique de programmation qui se concentre sur les données (les objets) et sur les interfaces avec ces objets. Pour faire une analogie avec la menuiserie, on pourrait dire qu'un menuisier "orienté objet " s'intéresse essentiellement à la chaise l'objet qu'il fabrique et non à sa conception (le "comment"). Par opposition, le menuisier "non orienté objet " penserait d'abord au "comment "...

- Simple

Nous avons voulu créer un système qui puisse être programmé simplement sans nécessiter un apprentissage ésotérique, et qui tire parti de l'expérience standard actuelle. En conséquence, même si nous pensions que C++ ne convenait pas, Java a été conçu de façon relativement proche de ce langage dans le dessein de faciliter la compréhension du système. De nombreuses fonctions compliquées, mal comprises, rarement utilisées de C++, qui nous semblaient par expérience apporter plus d'inconvénients que d'avantages, ont été supprimées de Java.

- Sécurité

Java a été conçu pour être exploité dans des environnements serveur et distribué. Dans ce but, la sécurité n'a pas été négligée. Java permet la construction de systèmes inaltérables et sans virus.

- Architecture neutre

Le compilateur génère un format de fichier objet dont l'architecture est neutre – le code compilé est exécutable sur de nombreux processeurs, à partir du moment où le système d'exécution de Java est présent. Pour ce faire, le compilateur Java génère des instructions en bytecode qui n'ont de lien avec aucune architecture particulière. Au contraire, ces instructions ont été conçues pour être à la fois faciles à interpréter et faciles à traduire en code natif.

- Portable

A la différence du C/C++, on ne trouve pas les aspects de dépendance de la mise en œuvre dans la spécification. Les tailles des types de données primaires sont spécifiées, ainsi que le comportement arithmétique qui leur est applicable.

- Interprété

L'interpréteur Java peut exécuter les bytecodes directement sur n'importe quelle machine sur laquelle il a été porté. Dans la mesure où la liaison est un processus plus incrémentiel et léger, le processus de développement peut se révéler plus rapide et exploratoire.

- Performances élevées

En général, les performances des bytecodes interprétés sont tout à fait suffisantes, il existe toutefois des situations dans lesquelles des performances plus élevées sont nécessaires. Les bytecodes peuvent être traduits à la volée en code machine pour l'unité centrale destinée à accueillir l'application.

- Multithread

Les avantages du multithread sont une meilleure interréactivité et un meilleur comportement en temps réel.

II.2.2 Choix de la technologie Web

Il est évident de contourner les technologies Web essentielles du marché, avant de déterminer notre choix à l'ISAE-Cnam Liban. Par ailleurs, les technologies Web les plus utilisées sont les suivantes :

- Famille Open Source: PHP, Jsp, Jsf, plateforme J2EE, EJB.
- Famille propriétaires, comme les produits du Microsoft : plateforme .NET, ASP.

Parmi ces multiples technologies Web, il fallait trouver la plus populaire et la plus utilisée dans des millions des sites Web pour assurer sa continuité, et qui respecte une architecture pour faciliter la maintenance et l'évolution de notre futur logiciel, celui ayant la meilleure rentabilité et surtout celui le plus approprié avec nos besoins.

II.2.2.1 JSP

JSP a l'avantage d'intéresser deux communautés différentes de développeurs. Les développeurs de sites Internet qui utilisent JSP cherchent à générer du contenu dynamique de façon efficace et rapide. Les programmeurs Java implémentent un code en arrière plan pour les applications qui traiteront toutes les données.

Présentation

Les pages web JSP sont une technologie développée par Sun (maintenant Oracle) basée sur Java qui simplifie le processus de développement de sites web dynamiques. L'utilisation du langage de programmation Java permet aux concepteurs de pages qui utilisent JSP d'incorporer rapidement des éléments dynamiques dans les pages web en intégrant du code Java et en utilisant quelques balises (tags) simples.

Ces balises fournissent aux développeurs HTML la possibilité d'utiliser la logique objet Java sans pour autant devoir maîtriser toute la complexité de développement de programmes Java.

JSP est un langage script côté serveur. Le code de la page est exécuté par le serveur web. Les pages JSP sont des fichiers texte munis de l'extension .jsp, qui remplacent les pages HTML traditionnelles. Les fichiers JSP contiennent un code HTML et un code imbriqué en Java.

Le mélange d'HTML et de code Java dans les pages JSP permet de séparer la présentation (en HTML) des aspects procéduraux contenus dans le code. On a ainsi une grande souplesse dans le développement de sites web.

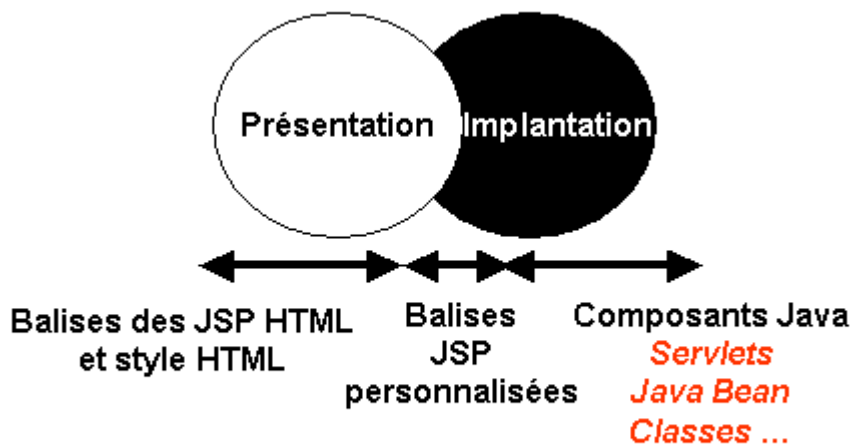


Figure 5: Presentation JSP

II.2.2.2 Autres langages dynamiques

CGI

Les premiers serveurs HTTP ne comportaient pas de programmes pour générer des réponses de façon dynamiques. Des interfaces étaient utilisées pour appeler d'autres programmes capables de traduire les requêtes en un contenu exécutable. Le premier standard utilisé a été CGI (Common Gateway Interface), c'est une norme qui définit un mécanisme qui permet au serveur HTTP de transmettre les informations d'une requête à des programmes externes. Le langage Perl est le langage le plus utilisé pour écrire des scripts CGI, mais on peut écrire des langages scripts dans bien d'autres langages.

L'inconvénient de ces scripts est qu'ils sont très gourmand en ressources systèmes. Chaque script utilise un processus différent ce qui demande beaucoup de mémoire et d'utilisation processeur.

ASP

L'équivalent de JSP chez Microsoft est ASP, il est exécuté du côté serveur et supporte divers langages scripts tels que PerlScript, Jscript, C# et VBScript. Le langage par défaut qui est utilisé est VBScript qui est un sous ensemble de Visual Basic. VBScript permet d'accéder aux composants ActiveX. Ces composants sont des codes objets compilés pouvant, dans la pratique, encapsuler toutes sortes de fonctionnalités telles que l'accès à une base de donnée ou à la manipulation de fichiers etc. L'inconvénient majeur de ASP est qu'il n'est disponible qu'avec un serveur de Microsoft (IIS) qui ne s'exécute que sous le système d'exploitation Windows.

PHP

Un des plus important concurrent de JSP est PHP créé en 1994. PHP est un langage interprété exécuté du côté serveur et non du côté client. La syntaxe du langage provient de celle du langage C, du Perl et de Java. Ses principaux atouts sont:

- La gratuité et la disponibilité du code source (PHP3 est distribué sous licence GNU GPL)
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, ...)
- La simplicité d'écriture de scripts
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML)
- La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés).

Exécution des scripts PHP par le serveur:

On se rend compte que l'interprétation et l'exécution d'un script est très coûteux pour le serveur. Un nombre important et simultané de requêtes peut ralentir fortement les temps de réponses.

II.2.2.3 Fonctionnement de JSP

JSP présente de nombreux avantages par rapport aux autres langages de génération de contenu. Comme il s'agit d'une technologie basée sur Java, JSP profite de tous ce qu'offre ce langage pour le développement et le déploiement d'applications. En effet Java est un langage orienté objet fortement typé, permettant l'encapsulation, le traitement des exceptions et la gestion automatique de la mémoire.

D'autre part, grâce aux API normalisées pour les JSP et à la portabilité du bytecode compilé Java, nous ne sommes pas limités à un seul type de plate-forme, de système d'exploitation. Nous pouvons changer les composants d'une page JSP comme nous le souhaitons.

De plus, JSP utilise l'ensemble de la plate forme JAVA sous-jacente ; par conséquent, les JSP peuvent directement tirer avantage de toutes les API Java.

JSP n'est pas rattaché à un éditeur particulier, les développeurs peuvent donc choisir les solutions de leurs choix.

Lorsque la page est demandée par un utilisateur en HTTP, alors le serveur web HTTP va transmettre sa requête à un moteur de JSP qui va interpréter la page, compiler le code et générer la réponse. L'utilisateur ne verra aucune ligne de code Java dans les sources de la page qu'il aura reçu.

Etapes d'une demande de page JSP:

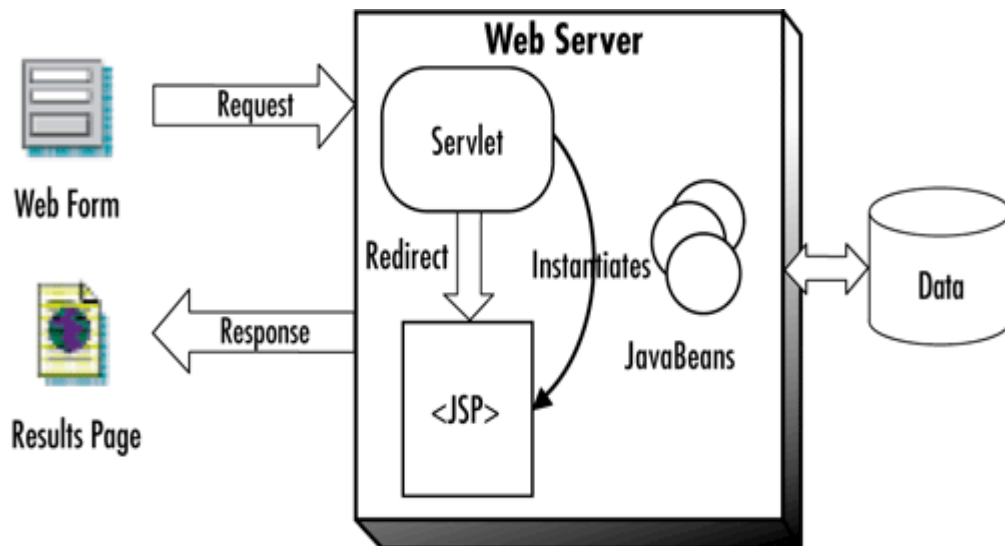


Figure 6: Etapes d'une demande de page JSP

II.2.2.4 JSP ou PHP ?

Les deux langages JSP et PHP sont relativement différents. Si vous débutez en programmation dynamique et que vous ne voulez pas perdre trop de temps à apprendre un langage de programmation, il est alors préférable de s'orienter vers PHP. C'est un langage simple qui offre une infinité de possibilités dans la création de pages web dynamiques.

PHP est plus adapté à la création de petits et moyens sites web qui ne sont pas trop complexes en terme de fonctionnalités.

JSP est une technologie Java, il faut donc une bonne connaissance de ce langage bien que des développeurs web puissent utiliser des balises JSP prédéfinies. Ce langage est bien adapté au développement de sites marchands ou de gros Intranets qui demandent la mise en place d'une architecture 3-tiers.[4]

De plus, du fait de l'utilisation de la *Virtual Machine Java*, JSP demandera un serveur plus puissant avec plus de mémoire pour avoir des temps de réponses rapides.

Le tableau ci-dessous présente les points importants de chacun de ces langages:

JSP	PHP
Utilise le langage Java qui est totalement objet	Langage procédurale avec la possibilité de faire de l'objet
Peut se connecter à tout type de bases de données	Peut se connecter à tout type de bases de données
Développent suivant une architecture (n-tiers), séparation entre présentation et code métier	pas d'architecture, pas de séparation entre présentation et code métier
Les temps de réponses sont équivalents	
Plus gourmand en ressources à cause de la JVM	Langage interprète à chaque appel

Table 2 : Comparaison entre JSP et PHP

En effet, l'ISAE-Cnam Liban a besoin d'un système capable de gérer plusieurs fonctionnalités, certains parmi eux sont complexes, par exemple l'inscription, l'admission et l'horaire. Le PHP n'est pas capable de satisfaire nos besoins au niveau d'architecture et de fonctionnalité, mais dans le but de construire un logiciel maintenable et évolutif c'est mieux d'adopter la technologie JSP/Java qui possède beaucoup de caractéristiques mentionnées ci-haut indispensables pour le SI de l'institut.

II.2.3 Choix de la base de données

II.2.3.1 Introduction

Avant de choisir une base de données il fallait connaître les bases de données opérable avec Java. En utilisant la technologie JDBC pour la connexion à la base de données, nous pouvons manipuler les informations à la base de données, comme la recherche, la modification et la suppression. Cette technologie est très importante pour des travaux qui sont exploités par JSP.

II.2.3.2 La présentation de JDBC (Java Database Connectivity)

En utilisant des balisages JSP ou Scriptlet pour faciliter la génération des contenus dynamiques, la logique des contenus est encapsulée dans des balisages et JavaBeans. La connexion aux bases de données est très importante pour des sites interactifs. Pour se connecter à une base de données, il a besoin de la technologie JDBC et ODBC. Mais le ODBC a les inconvénients suivants :

- 1) ODBC est une API fournie avec langage C, il y aura des problèmes de sécurité et d'intégrité.
- 2) En utilisant ODBC, il n'est pas possible de réaliser exactement de transférer les codes en langage C au Java API, car il n'y a pas de pointeur en langage JAVA, mais il a besoin d'utiliser le pointeur en utilisant l'ODBC.

Par rapport aux inconvénients d'ODBC, ce sera mieux de choisir la technologie JDBC pour connexion aux bases de données. Les JDBC constituent un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données. JDBC permet l'accès à des bases de données relationnelles à partir du langage Java. Le pilote JDBC est offert dans la plupart des systèmes des bases de données. Les programmes Java peuvent être connectés aux bases de données par ces pilotes, pour réaliser les opérations des données. Il y a aussi le pont JDBC-ODBC qui est aussi exploité par SUN, grâce à qui, le programme Java peut visiter de bases de données qui ont le pilote d'ODBC. JDBC est un JAVA API, qui peut exécuter des requêtes SQL. Il est très simple d'envoyer des requêtes SQL aux bases de données en utilisant le JDBC.

Grâce à JDBC on peut se connecter au base de données suivantes :

- MySQL
- PostgreSQL
- Ingres
- Firebird
- Zodb
- Oracle
- SQL Server
- Etc..

II.2.3.3 Comparaison entre MySQL, PostgreSQL, Ingres, Firebird et Zodb

La communauté « 01net. » pour l'actualité informatique et les nouvelles technologies sur internet et « Novaforge » le site communautaire géré par des experts *open source*, a comparé avec l'aide d'un jury d'experts, issus de sociétés de services et d'intégration, cinq bases de données open source. Le meilleur choix pour l'entreprise est PostgreSQL, suivi de MySQL.[6]

Par ailleurs, ce comparatif de bases de données *open source*, le premier du genre, a fait appel à un jury de cinq experts indépendants, l'objectif est de fournir une grille de comparaison aussi objective et détaillée que possible pour les cinq solutions jugées les plus intéressantes en entreprise. Par suite je voudrais expliquer le choix de MySQL à l'institut.

Le choix s'est tout d'abord tourné vers MySQL et PostgreSQL, qui sont les deux bases de données *open source* les plus déployées et qui présentent des communautés particulièrement larges et actives.

Le comparatif a également intégré Ingres et Firebird. Elles n'ont pas la même notoriété que les deux précédentes. A l'origine, ces deux solutions étaient propriétaires et n'ont été “ libérées ” que depuis quelques années. Elles disposent, sous leurs formes commerciales, de parcs de déploiement assez importants. Elles vont donc assurément prendre leur place sur l'échiquier des bases de données *open source*.

Cette comparaison a ensuite retenu Zodb. Contrairement aux autres solutions du comparatif, celle-ci n'est pas une base relationnelle, mais orientée objet. Elle est fortement liée au langage de programmation Python et déployée le plus souvent au sein du serveur d'application Zope. Cette base permet de donner un éclairage différent sur ce comparatif.

Evolution des SGBD *open source*

D'abord utilisés dans le cadre universitaire, puis par des acteurs d'Internet, les systèmes de gestion de bases de données *open source* ont réussi à prendre place au cœur du système d'information des entreprises. Ils ont prouvé leur fiabilité et leur capacité à tenir la charge et font preuve d'un niveau de maturité et d'une couverture fonctionnelle satisfaisant.

Désormais, ils sont également utilisées au sein d'applications critiques ou métiers, y compris chez les grands comptes. Les directeurs informatiques ne font plus de différence hiérarchique entre bases propriétaires et bases *open source*. Ils font leur choix en fonction des besoins de leur projet.

Les PME profitent des faibles coûts d'acquisition

Parmi les solutions existantes sur le marché, les bases MySQL et PostgreSQL ont connu un développement particulièrement rapide. Certains éditeurs propriétaires comme Ingres ou Firebird ont profité de cette dynamique pour se positionner comme solution *open source*. D'une manière générale, grâce à leur faible coût d'acquisition et à leur grande accessibilité, les bases de données *open source* ont permis de démocratiser ce type de logiciel d'infrastructure, notamment auprès des PME.

Toutefois, dans certains domaines, les bases de données propriétaires ont encore une longueur d'avance, comme la réplication, la haute disponibilité, le partitionnement des données et la sauvegarde. Il est à noter que la migration d'une base propriétaire vers une solution *open source* ne doit pas être prise à la légère, car chaque solution a ses spécificités (langage procédural, performances, fonctionnalités, etc.). Il faut impérativement prévoir des phases d'études et de prototypages.

Le comparatif détaillé

Voilà le comparatif qui évalue la meilleure base de données en basant des critères bien définis, Pour chaque critère, la note de 1 à 5 :

notes sur 5	MySQL	PostgreSQL	Ingres	Firebird	Zodb
Utilisateurs	5*	5	2,5	2	3
Communauté	4,5	5	2	2,5	3,5
Offres commerciales	5	4	3,5	2,5	3,5
Fonctionnalités	4	5	4	4	4
Robustesse	4	5	4	4	4
Performance	4,5	4,5	4	4	3,5
Documentation	4,5	5	4	2	2
Administration	4	4,5	3	3	3
Utilisation	4,5	5	2,5	3	3
Déploiement	4,5	5	3	3,5	4
Note générale	4,5	5	3	3	3,5

Table 3 : Comparatif détaillé entre MySQL, PostgreSQL, Ingres, Firebird et Zodb

Les critères d'évaluation :

- **Utilisateurs**, pour examiner la taille de la base d'utilisateurs;
- **Communauté**, pour juger du dynamisme et du nombre de contributeurs au code ;
- **Offres commerciales**, pour estimer la qualité et l'abondance des services de support et des offres de formation ;
- **Fonctionnalités**, pour évaluer l'étendue des fonctions de la solution ;
- **Robustesse**, pour jauger l'aptitude de la solution à être utilisée au sein d'un environnement de production, voire au sein d'une architecture critique ;
- **Performance**, pour estimer les performances techniques de la solution (vitesse de traitement, requêtes par seconde, etc.) ;
- **Documentation**, pour apprécier l'étendue et le niveau de détail de la documentation ;
- **Administration**, pour évaluer les outils mis à la disposition des administrateurs ;
- **Utilisateur**, pour mesurer le confort d'utilisation pour le développeur d'application ;
- **Déploiement**, pour juger de la facilité et de la rapidité de déploiement.

Voilà un tableau qui représente quelques caractéristiques essentielles des SGBD déjà cités, et qui compare point par point quelques fonctionnalités les plus fréquemment utilisées de MySQL, PostgreSQL, Ingres, Firebird, Zodb .

	MySQL	PostgreSQL	Ingres	Firebird	Zodb
Environnements supportés	Linux, Windows, FreeBSD, Solaris, HP UX, AIX etc.	Linux, Windows, HP UX, FreeBSD, AIX, OS X, Unixware, Netware, etc.	Windows, Linux, AIX	Windows, Linux	Windows, Linux
Communauté	Oracle	Fujitsi, Pervasive Software, Bull	Des contributeurs	IB Phoeenix	ZodB
Réplication	Oui	Oui	Oui	Oui	Oui
SGBD Relationnel	Oui	Non	Oui	Oui	Non
SGBD Relationnel-Objet	Non	Oui	Non	Non	Non
SGBD Orienté Objet	Non	Non	Non	Non	Oui
Héritage	Non	Oui	Non	Non	Oui
Documentation	Anglais, Français	Anglais	Anglais	Anglais, Français	Succincte
Facilité d'emploi	Oui	Oui	Oui	Oui	Difficile
Transactions	Oui	Oui	Oui	Oui	Oui

Table 4: Comparaison entre MySQL, PostgreSQL, Ingres, Firebird et Zodb

II.2.3.4 Justification du Choix MySQL à l'institut

Selon le comparatif ci-dessus, la base de données PostgreSQL est la meilleure, suivie par MySQL. Pourtant, MySQL présente des avantages n'ont pas été cités dans ce comparatif, par exemple, MySQL est moins gourmand en ressource et plus rapide et actif que PostgreSQL. Par déduction, MySQL était la solution « idéale » à nos besoins, surtout en terme matérielles, car nous ne possédons pas des équipements assez performants et rapides (RAM du serveur 0.5 GHZ !). Donc nous voudrions un système qui utilise le minimum de ressources et qui satisfait le maximum de besoins, or c'est le cas du MySQL.

III Environnement méthodologique

III.1 Conduite de projet

III.1.1 Besoin de la méthodologie

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes de développement. Ces dernières ont toujours essayé d'apporter un contrôle continu sur un projet tout au long de son processus de vie. Bien que des méthodes de développement existent depuis 30 ans (Merise etc.), nous ne pouvons constater aujourd'hui l'existence d'une règle qui soit à la fois formelle et commune aux différentes cultures. Par ailleurs, des méthodes séquentielles comme celles se basant sur le cycle en V, ont montré leur limite dans un environnement régi par des changements réguliers, impliquant une impossibilité à revenir en arrière, et de ce fait, laissant une très petite marge à l'erreur. Avec l'avènement de la pensée objet, de nouvelles méthodes sont apparues et différentes notations ont été établies. UML a ouvert le terrain de l'unification en fusionnant ces notations et en apportant précision et rigueur à la définition des concepts introduits. Sur cet élan, les spécialistes ont aussi pensé à unifier non pas les processus, mais plus exactement les meilleures pratiques de développement orienté objet.

Le but du chapitre est d'appliquer une méthode rigoureuse de conduite d'un projet réel. Le projet en question concerne la gestion de l'ISAE-Cnam Liban, qui englobe beaucoup des fonctionnalités. Nous espérons à travers ce chapitre, démontrer l'importance de l'application d'une méthodologie de développement, et aussi permettre par la suite que ce produit puisse être évolutif et facilement maintenable par des intervenants tiers.

III.1.2 Évolution des approches de développement

Par le passé, le modèle Entité-Relation représentait une grande partie des approches les plus utilisées. Actuellement, les nouvelles technologies s'appuient sur le modèle objet. En termes d'analyse et de modélisation objet, UML est devenu un standard incontournable, stabilisé, industriel.

Au début, le cycle en cascade (ex : le cycle en V) était très utilisé. Mais on a vite constaté son incapacité à s'adapter aux différents changements. Une méthode semi-itérative est apparue (RAD) pour pallier aux carences de ce dernier. L'itératif s'est ensuite imposé, parce qu'il réduit la complexité de réalisation des phases, en travaillant par approches successives et incrémentales. Une méthode fortement axée sur l'itératif et le modèle UML est alors apparue, il s'agit de UP (Unified Process). Cette méthode comme son nom l'indique, a été le fruit de travail de plusieurs personnes voulants « unifier » les différentes méthodes objets existantes à ce moment comme Booch, OMT. On constate aujourd'hui, l'émergence d'une nouvelle approche : les méthodes agiles (Agile Modeling). C'est des méthodes itératives à planification souple

qui leur permettent de s'adapter à la fois aux changements du contexte et de spécifications du projet.

III.1.3 Nécessité d'une méthode de développement

De nos jours, une des tendances les plus en vue et qui concerne tout les secteurs de développement, est l'informatisation. Depuis l'apparition de l'informatique et son introduction dans le monde économique, les entreprises et entités publiques aspirent à optimiser et à rendre fiable la gestion de leur structure interne. L'ISAE-Cnam Liban possède quelques milliers d'étudiants qu'il est difficile de gérer en continu. Avec la mise en place récente du système LMD et le ECTS, la situation s'est davantage compliquée et les tâches de gestion sont devenues plus complexes. Pour cela, l'institut a commencé de développer son propre SI, qui a comme but de faciliter sa gestion. Mais le volume des tâches et leurs complexités, nous a obligé d'adopter une méthode de développement fiable que je veux la traiter au paragraphe suivant.

III.1.4 La méthode 2TUP

Devant le nombre de méthodes disponibles, le choix parmi elles devient difficile, beaucoup de questions peuvent se poser à un chef de projet lors d'un démarrage de projet :

- Comment pourrai-je organiser les équipes de développement ?
- Quelles tâches attribuer à qui ?
- Quel temps faudra-il pour livrer le produit ?
- Comment faire participer le client au développement afin de capter les besoins de celui-ci ?
- Comment éviter des dérives et de mauvaises estimations qui vont allonger les coûts et le temps de développement ?
- Comment voudrai-je procéder pour que le produit soit évolutif et facilement maintenable ?

Nous pouvons citer à ce propos les méthodes de développement objet suivantes : 2TUP, XP. Notre choix s'est porté vers la méthode 2TUP, du fait de son approche nouvelle, originale. Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final. Ce processus se base lui-même sur le Processus Unifié (Unified Process) qui est devenu un standard général réunissant les meilleures pratiques de développement. Cette méthode ne se base aucunement sur un processus linéaire mais bien, sur un développement itératif et incrémental. Je voudrai d'abord définir les différents concepts qui vont être utilisés dans mon mémoire d'ingénieur.

III.1.4.1 Définition d'un processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. (Rocques & Vallée, 2004)

III.1.4.2 Le processus unifié

Le Processus Unifié (PU ou UP en anglais pour Unified Process) est une méthode de développement logiciel construite sur UML ; elle est itérative et incrémentale, centrée sur l'architecture, conduite par les cas d'utilisation et pilotée par les risques.

- itérative et incrémentale : la méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir aussi à ajouter de nouveaux incréments.
- conduite par les cas d'utilisation : elle est orientée utilisateur pour répondre aux besoins de celui-ci.
- centrée sur l'architecture : les modèles définissent tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.
- pilotée par les risques : en définissant des priorités pour chaque fonctionnalité, on peut minimiser les risques d'échec du projet.

La gestion d'un tel processus est organisée d'après les 4 phases suivantes :

- Prétude : c'est ici qu'on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité).
- Elaboration : sert à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base.
- Construction : sert à livrer progressivement toutes les fonctions du système.
- Transition : déployer le système sur des sites opérationnels.

Chaque phase est elle-même décomposée séquentiellement en itérations limitées dans le temps (entre 2 et 4 semaines). Le résultat de chacune d'elles est un système testé, intégré et exécutable. L'approche itérative est fondée sur la croissance et l'affinement successifs d'un système par le biais d'itérations multiples. Le système croît avec le temps de façon incrémentale, itération par itération, et c'est pourquoi cette méthode porte également le nom de développement

itératif et incrémental. Il s'agit là du principe le plus important du Processus Unifié.

Ces activités de développement sont définies par 6 disciplines fondamentales qui décrivent la capture des besoins, la modélisation métier, l'analyse et la conception, l'implémentation, le test et le déploiement.

Notons que ces différentes étapes (ou disciplines) peuvent se dérouler à travers plusieurs phases. Le processus unifié doit donc être compris comme une trame commune des meilleures pratiques de développement.

III.1.4.3 Le processus 2TUP adopté à l'institut

On dit de la méthode UP qu'elle est générique c.à.d. qu'elle définit un certain nombre de critères de développement, que chaque société peut par la suite personnaliser afin de créer son propre processus plus adapté à ses besoins. C'est dans ce cadre que la société Valtech a créé la méthode 2TUP. 2TUP signifie « 2 Track Unified Process ». C'est un processus qui répond aux caractéristiques du Processus Unifié. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.



Figure 7: Le système d'information soumis à deux types de contraintes

La branche gauche (fonctionnelle) : capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d'information sont en effet indépendantes des technologies utilisées. Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs.
- L'analyse.

La branche droite (architecture technique) : capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme.

Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser.

Cette branche comporte les étapes suivantes :

- La capture des besoins techniques.
- La conception générique.

La branche du milieu : à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de Y.

Cette branche comporte les étapes suivantes :

- La conception préliminaire.
- La conception détaillée.
- Le codage.
- L'intégration.

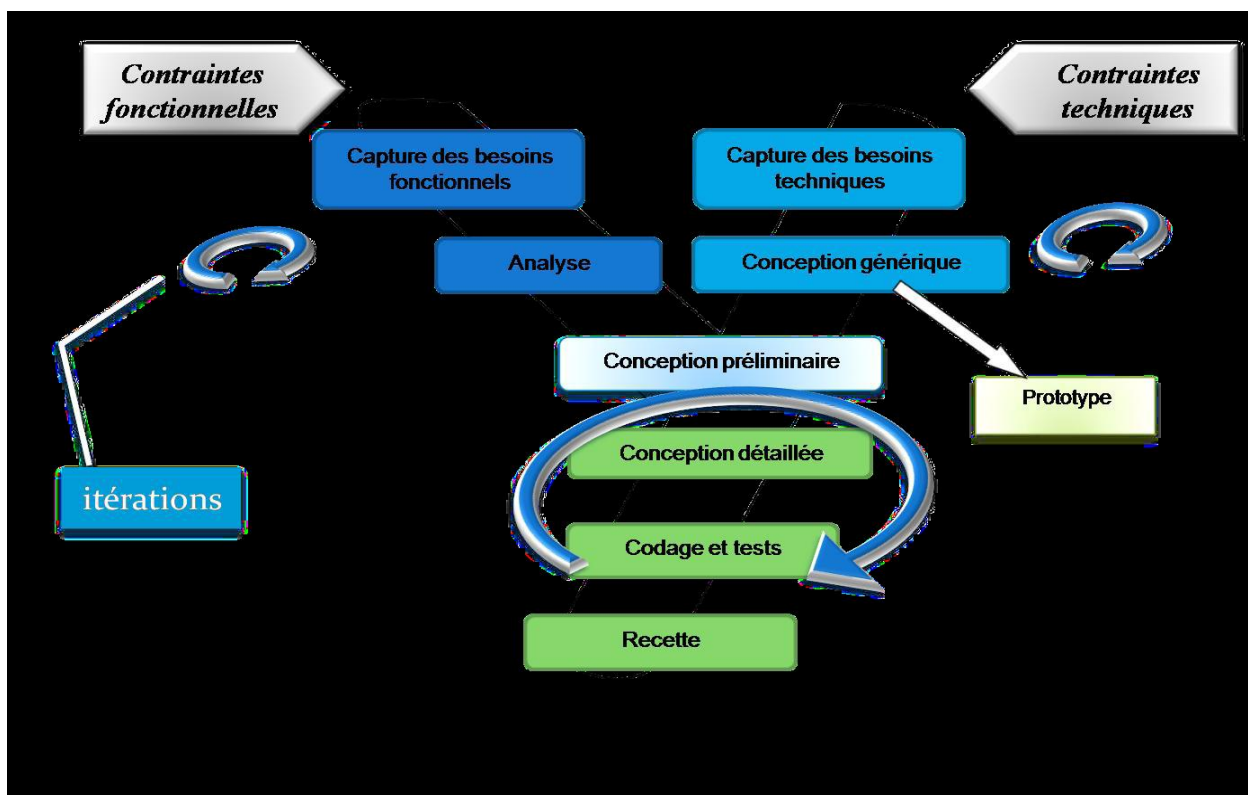


Figure 8: Le processus de développement en Y

III.1.4.4 Un processus de modélisation avec UML

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leur facilité et clarté, de bien modéliser le système à chaque étape.

« Unified Modeling Language » : UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solu-

tions et communiquer des points de vue (Pitman, 2006).

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est absolument pas.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage (Roques, 2006).

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet :

- Le diagramme des cas d'utilisation : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.
- Le diagramme d'activités : représente les règles d'enchaînement des activités et actions dans le système. Il peut être assimilé comme un algorithme mais schématisé.
- Le diagramme de packages : présent depuis UML 2.0, ce diagramme modélise des catégories cohérentes entre elles, pour un souci de partage des rôles. Correspond à l'étape de modélisation des différents scénarios d'un cas d'utilisation.
- Le diagramme de classes : sûrement l'un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet.
- Le diagramme de séquence : représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système.
- Le diagramme d'états : représente le cycle de vie d'un objet. Il spécifie les états possibles d'une classe et leur enchaînement.

Ce diagramme est utilisé lors des étapes d'analyse et de conception.

III.2 Programmation Orienté Objet (POO)

III.2.1 Définition

La programmation orientée objet (POO), ou programmation par objet, a été élaborée par Alan Kay dans les années 1970. C'est un paradigme de programmation informatique qui consiste en la définition et l'interaction de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait communiquer avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; la communication entre les objets via leur relation permet de réaliser les fonctionnalités attendues, de résoudre le ou les problèmes.[7]

Orthogonalement à la programmation par objet, afin de faciliter le processus d'élaboration d'un programme, existent des méthodologies de développement logiciel objet dont la plus connue

est USDP (*Unified Software Development Process*).

Il est possible de concevoir par objet une application informatique sans pour autant utiliser des outils dédiés. Il n'en demeure pas moins que ces derniers facilitent de beaucoup la conception, la maintenance, et la productivité. On en distingue plusieurs sortes :

- les langages de programmation (Java, C#, C, Eiffel, Python, Ruby, C++, PHP, Smalltalk etc.)
- les outils de modélisation qui permettent de concevoir sous forme de schémas semi-formels la structure d'un programme (Objecteering, UMLDraw, Rhapsody, DBDesigner etc.)
- les bus distribués (DCOM, CORBA, RMI, etc.)
- les ateliers de génie logiciels ou AGL (WinDev et son langage le Wlangage)

III.2.2 Historique

Le langage Simula-67, en implantant les Record Class de Hoare, pose les constructions qui seront celles des langages orienté objet à classes : classe, polymorphisme, héritage, etc. Mais c'est réellement par et avec Smalltalk 71 puis Smalltalk 80 (Dan Ingalls), inspiré en grande partie de Simula 67 et de Lisp, que les principes de la programmation par objets, résultat des travaux d'Alan Kay, sont véhiculés : objet, encapsulation, messages, typage et polymorphisme (via la sous-classification) ; les autres principes, comme l'héritage, sont soit dérivés de ceux-ci ou une implantation. Dans Smalltalk, tout est objet, même les classes. Il est aussi plus qu'un langage à objets, c'est un environnement graphique interactif complet.

À partir des années 1980, commence l'effervescence des langages à objets : Objective C (début des années 1980), C++ (C avec classes) en 1983, Eiffel en 1984, Common Lisp Object System dans les années 1980, etc. Les années 1990 voient l'âge d'or de l'extension de la programmation par objet dans les différents secteurs du développement logiciel.

Depuis, la programmation par objet n'a cessé d'évoluer aussi bien dans son aspect théorique que pratique et différents métiers et discours mercatiques à son sujet ont vu le jour :

- l'analyse objet (AOO ou OOA en anglais),
- la conception objet (COO ou OOD en anglais),
- les bases de données objet (SGBDOO),
- les langages objets avec les langages à prototypes,
- ou encore la méthodologie avec MDA (Model Driven Architecture).

Aujourd'hui, la programmation par objet est vue davantage comme un paradigme, le paradigme objet, que comme une simple technique de programmation. C'est pourquoi, lorsque l'on parle de nos jours de programmation par objets, on désigne avant tout la partie codage d'un modèle à objets obtenu par AOO et COO.

III.2.3 Modélisation objet

La modélisation objet consiste à créer un modèle informatique du système de l'utilisateur (un système informatique). Ce modèle peut rassembler aussi bien des éléments du monde réel que

des concepts ou des idées propres au métier ou au domaine duquel fera partie le système. La modélisation Objet consiste à définir, à qualifier dans un premier temps ces éléments sous forme de types, donc indépendamment de la mise en œuvre. C'est ce que l'on appelle l'analyse orientée objet ou OOA (Object-Oriented Analysis).

Puis, on propose une ou des solutions techniques pour représenter les éléments définis dans le système informatique. C'est ce que l'on appelle la conception orientée objet ou OOD (Object-Oriented Design). Une fois un modèle de conception établi, il est possible au développeur de leur donner corps dans un langage de programmation. C'est ce que l'on appelle la programmation orientée objet ou OOP (Object-Oriented Programming). À un modèle d'analyse peuvent correspondre plusieurs modèles de conception.

Pour écrire ces différents modèles, différents langages et méthodes ont été mis au point, dont OMT (Object Modeling technique) de Rumbaugh, BOOCH'93 de Booch et OOSE de Jacobson. Toutefois, ces méthodes ne permettaient de modéliser que certains types d'applications et se trouvaient limitées dans d'autres contextes. La méthode OMT prévalait sur l'ensemble des autres méthodes dans la première partie de la décennie 1990.

À partir de 1994, Rumbaugh, Booch et Jacobson ont décidé de s'unir dans l'élaboration d'une nouvelle méthode, suffisamment générique, pour pouvoir s'appliquer à quasiment tous les contextes applicatifs. Ils ont commencé d'abord par définir un langage de modélisation fortement inspiré de celles des méthodes des trois auteurs : UML. Une fois celui-ci pris en charge par l'OMG (Object Management Group), un organisme destiné à standardiser des technologies objet, comme CORBA (Common Object Request Broker Architecture), un intergiciel (middleware en anglais) objet réparti, Rumbaugh, Booch et Jacobson se sont attaqués à la méthode proprement dite: USDP. Cette méthode définit un cadre générique de développement objet avec UML comme langage de modélisation. USDP (généralement raccourci en UP) est une méthode itérative et incrémentale, centrée sur l'architecture et guidée par les cas d'utilisation et la réduction des risques. C'est aux concepteurs de s'attribuer cette méthode en l'instanciant à leur métier et à leur domaine.

Néanmoins pour un certain nombre de concepteurs objet, dont Bertrand Meyer, l'inventeur du langage orienté objet Eiffel, guider une modélisation objet par des cas d'utilisations est une erreur de méthode qui n'a rien d'objet et qui est plus proche d'une méthode fonctionnelle. Pour eux, les cas d'utilisations sont relégués à des utilisations plutôt annexes comme la validation d'un modèle par exemple.[12]

III.3 Limitations de la programmation orienté objet

La POO est une approche générale de la conception d'un logiciel mais elle n'offre pas une solution technique comme la manière de codage et d'architecture etc. bien sûr les développeurs ont besoin des bases scientifiques et méthodologiques pour bien développer et concevoir les besoins, d'autre part on peut poser ces questions suivantes :

- Y a-t-il des méthodes pour réarranger et organiser le code, pour améliorer la maintenance et la lisibilité?
- Y a-t-il des méthodes au niveau du code et non seulement des approches générales, afin d'aider les programmeurs et les concepteurs?

Donc ces limitations nous pousse comme développeur et analyseur à l'ISAE-Cnam Liban à chercher des nouvelles méthodes plus détaillées, dans ce cadre se situe le chapitre suivant : la programmation orienté composant.

III.4 Programmation Orientée Composant (POC)

III.4.1 Introduction

De nos jours tous les programmeurs savent et pratiquent intensivement la POO C'est une méthode de conception effectivement très mature, il est possible de mal l'utiliser (mauvais choix de granularités des objets, relations mal définies), mais armé d'une bonne analyse tout doit bien se passer.[8]

La POO à sans doute été un pas en avant énorme, les programmes sont tout de suite beaucoup mieux structurés et compréhensibles si les objets sont correctement choisis.

III.4.2 Définition

Un objet c'est bien, mais que diriez vous d'un objet boîte noire pour lequel vous ne verriez que un port de connexion (une interface). Ce concept est appelé Programmation Orientée Composant,

En effet un objet muni d'une interface est appelé un composant.

Pour bien comprendre ce principe de composant il vous suffit d'imaginer que vous créez un port matériel bien défini et sur lequel vous vous êtes mis d'accord avec les utilisateurs avant toute conception du composant en elle même. Les représentations graphiques des interfaces (en analyse) rappellent d'ailleurs ce concept. Ce port permet aux utilisateurs de se connecter à votre boîte noire sans en connaître le contenu.

Donc la POC consiste à utiliser une approche modulaire au niveau de l'architecture d'un projet informatique, ce qui permet d'assurer au logiciel une meilleure lisibilité et une meilleure maintenance. Les développeurs, au lieu de créer un exécutable monolithique, se servent de briques réutilisables.

La POC n'est pas sans similitudes avec la POO, puisqu'elle revient à utiliser une approche objet, non pas au sein du code, mais au niveau de l'architecture générale du logiciel.

La POC est particulièrement pratique pour le travail en équipe et permet d'industrialiser la création de logiciels.

III.4.3 Structure d'un composant

Comme le composant n'est utilisable qu'une fois compilé, la méthode d'implémentation importe peu. Il est possible d'utiliser une approche objet ou procédurale. Il convient d'éviter une

confusion courante : la POC et la POO sont des choses différentes. Utiliser la POO pour développer un composant est facultatif. En fait, l'implémentation du composant n'a aucune influence sur les programmes clients, du moment que les signatures de l'interface ne changent pas.

Le code d'un composant peut en effet être séparé en deux parties. Tout d'abord, les méthodes et données internes. Un composant peut implémenter des fonctions qu'il utilise « pour son compte personnel », et qui ne sont pas accessibles de l'extérieur. On parle de méthodes « privées ». Ensuite, le composant, pour pouvoir être utilisé, doit fournir un moyen de communication avec les programmes clients. Certaines fonctions sont donc accessibles de l'extérieur, et dévolues à être appelées par ces programmes. On parle de méthodes « publiques », ou d'« interface ».

III.4.4 Niveaux d'abstraction du POC

Une conception orientée composants à différents niveaux d'abstraction peut être schématisée comme illustré par la figure 9.

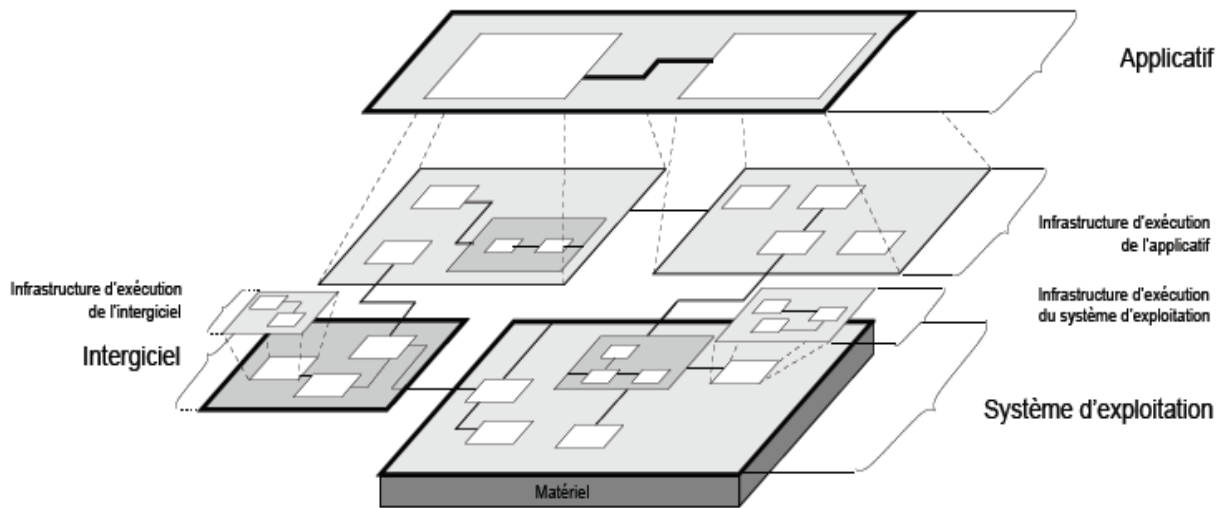


Figure 9: Paradigme composant à différents niveaux d'abstraction d'un système

III.4.5 Les avantages à utiliser la POC

Les avantages à utiliser une approche POC, pour conduire un projet sont multiples :

- spécialisation : L'équipe de développement peut-être divisée en sous-groupes, chacun se spécialisant dans le développement d'un composant
- sous traitante : Le développement d'un composant peut-être externalisé, à condition d'en avoir bien réalisé les spécifications au préalable

- facilité de mise à jour : La modification d'un composant ne nécessite pas la recompilation du projet complet
- facilité de livraison/déploiement : Dans le cas d'une mise à jour, d'un correctif de sécurité, alors que le logiciel a déjà été livré au client, la livraison en est facilitée, puisqu'il n'y a pas besoin de re-livrer l'intégralité du projet, mais seulement le composant modifié
- choix des langages de développement : Il est possible, dans la plupart des cas, de développer les différents composants du logiciel dans des langages de programmation différents. Ainsi, un composant nécessitant une fonctionnalité particulière pourra profiter de la puissance d'un langage dans un domaine particulier, sans que cela n'influe le développement de l'ensemble du projet
- productivité : La réutilisabilité d'un composant permet un gain de productivité non négligeable car elle diminue le temps de développement, d'autant plus que le composant est réutilisé souvent.[13]

III.4.6 Conclusion

Bien que l'utilisation de la POC soit réellement appréciable dans la conduite d'un projet de développement, elle n'est pas sans imposer quelques contraintes importantes.

Tout d'abord, la POC est une méthode dont le bénéfice se voit surtout sur le long terme. En effet, lorsque l'on parle de réutilisation, de facilité de déploiement, c'est que le développement est sinon achevé, du moins bien entamé. Mais factoriser un logiciel en composants nécessite un important travail d'analyse. La rédaction des signatures des méthodes devra être particulièrement soignée, car modifier une signature nécessitera de retravailler toutes les portions de codes du projet qui font appel au composant, et l'on perdrait alors les bénéfices de l'indépendance des briques logicielles.

En un mot, si la POC industrialise le développement, la phase de conception du logiciel prendra un rôle encore plus important.

Le fait de ne pas connaître l'implémentation d'un composant (à moins d'avoir accès au source), peut également gêner certains chefs de projets qui veulent garder un contrôle total sur leur logiciel.

Et encore plus, en poussant ce principe plus loin, deux autres avantages apparaissent :

- La possibilité de distribuer physiquement facilement ses modules, , par exemple la technologie DCOM (Distributed Component Object Model) de Microsoft permet de faire appel à des composants situés sur un réseau, et CORBA(Common Object Request Broker Architecture) de OMG.
- La possibilité de mettre en place de meilleures méthodologies de tests. Vous pouvez créer des Modules de tests qui viendront se brancher sur vos interfaces afin de tester les réactions de votre composant. S'ouvrent alors à vous les joies du « Unit Test ».

Par ailleurs, il est vrai que l'utilisation réelle de POC n'est pas triviale. Définir les interfaces qui seront implémentées avant même l'écriture des blocs métier peut paraître ardue. Je pense

qu'il faut une certaine expérience de génie logiciel avant d'être capable de réaliser dès l'analyse des versions définitives d'interface. C'est cependant un travail qui doit porter ses fruits, et puis le temps passé sur l'analyse n'est jamais perdu.

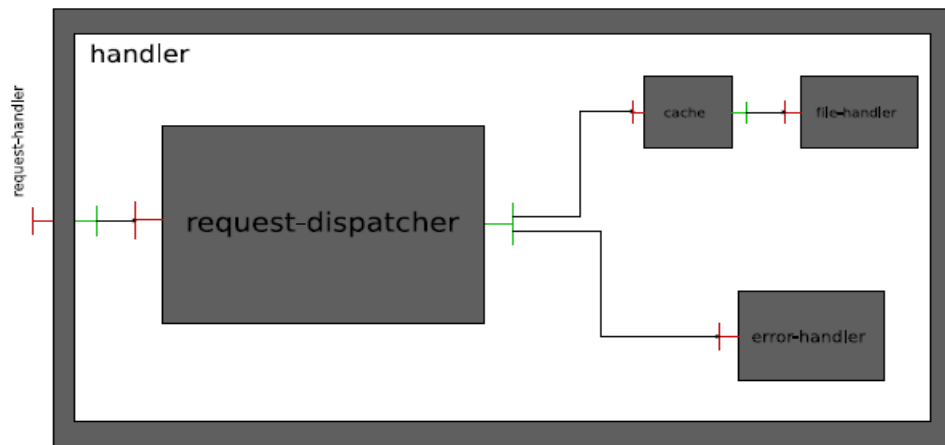


Figure 10: Exemple d'un modèle composant

Voilà un modèle de composant dans le SI de l'institut qui montre que les trois composants (horaire-professeurs, horaire-chef et horaire-étudiant) utilisent le même composants qui s'appelle «interface horaire » en passant les paramètres convenables (numéro professeurs etc.) :

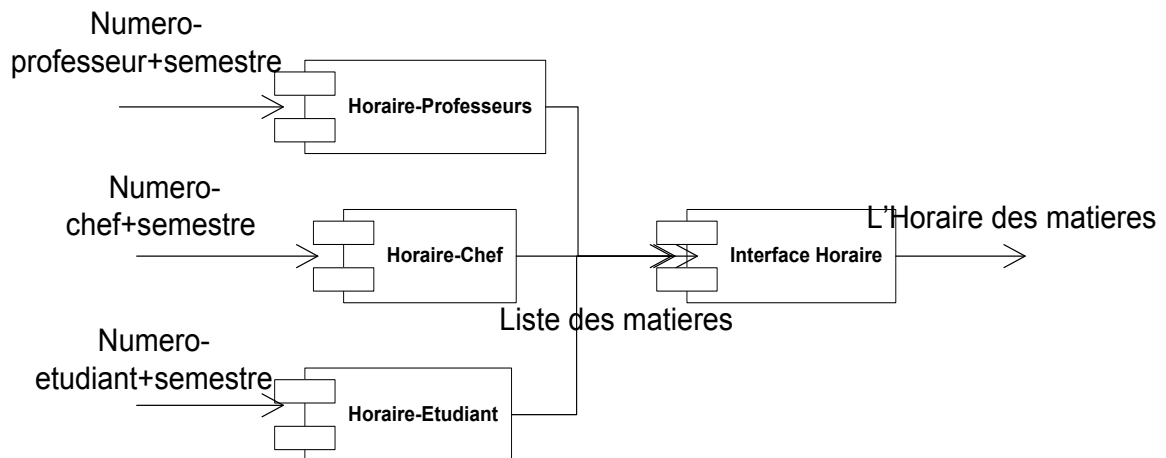


Figure 11: Exemple d'un modèle composant implémenté à l'institut

IV Analyse et conception

IV.1 Introduction

Modéliser un Système d'Information d'une universités est un travail fastidieux compte tenu de la complexité du système et de nombres important d'acteurs qu'il peut impliquer.

Toutefois, pour notre projet, nous avons procéder à quelques simplifications, afin de pouvoir mettre en évidence l'apport de la modélisation UML pour la mise en place d'Architecture Orientés Services.

L'objectif d'une architecture orientée services est donc de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.

L'idée sous-jacente est de cesser de construire la vie de l'entreprise autour d'application, pour faire en sorte de construire une architecture logicielle globale décomposées en services correspondant aux processus métiers de l'entreprise.

IV.2 Analyse

Une meilleur analyse de besoins abouti à une conception robuste, par suite à un logiciel flexible qui satisfait les besoins. Les besoins préliminaires sont déjà cités dans le chapitre I « Environnement du travail-Section I.4 Nouveaux besoins ». Beaucoup des activités sont réalisées avant l'établissement des services essentiels du SI, citons à titre d'exemple :

- L'analyse des failles de l'ancien SI (voir Chapitre I- section I.3) .
- Faire des réunions périodiques avec le directeur, les chefs de départements et des cellules, les comptables, et les chefs des centres, afin de partager nos opinions pour comprendre mieux les besoins et vérifier les idées essentiels.

En effet, je ne voudrai pas expliquer toutes les points du cahier de charges, mais seulement les essentiels, et encadrer les suivantes :

- Analyse de l'existant
- Conserver les bases du statut
- Intégrer des nouvelles notions
- Résoudre les anciennes fautes
- Problèmes rencontrés

IV.2.1 Analyse de l'existant

Les modules informatisés basant sur l'ancien SI sont :

- La gestion des étudiants.
- La gestion des enseignants.
- La gestion du stock.

En effet, il faut élargir le scope du nouveau SI pour pouvoir encadrer la gestion des ressources humaines, la comptabilité et la bibliothèque. En plus il faut réétudier les trois modules cités ci-haut du nouveau, car ils possèdent beaucoup des points faibles et des conflits dont je voudrai expliquer ultérieurement.

IV.2.2 Conserver les bases du statut

Les notions de bases du statut de l'institut qui caractérise l'ancien système sont conservé dans la nouvelle analyse et dans le nouveau cahier des charges, car l'institut n'a pas changé sa mission générale, mais le changement réside dans la structure administrative surtout après l'accroissement géographique de l'institut et l'introduction du notion ECTS depuis 2005. Le respect de certains de certain normes était important lors de l'établissement du cahier des charges, rappelons que la direction qui décide, et tous les normes subies des changement, citons à titre d'exemple :

- Résultats avant l'année 2005 [11]
 - Une composition subissait deux corrections par deux correcteurs différents. En cas de différence dépassant deux notes, une délibération aurait lieu.
 - La note d'une composition était faite sur vingt.
 - Le calcul de la note finale était fait sur 40, avec une pondération de 50% pour chaque composition. Parfois, la pondération se changeait à l'aide de coefficients pour 1^{ère} et 2^{ème} composition.
 - En général, la note de réussite par composition était 10. Cette note de réussite pouvait se changer pour devenir parfois 12.
 - Pour chaque résultat, il fallait mentionner les noms des professeurs correcteurs.
 - En général, les mentions étaient fixes ; mais avant l'année 2000, le groupe des matières formant le cycle C était évalué à l'aide des mentions différentes des mentions ordinaires et le calcul de la note finale se faisait avec une pondération pour chaque matière du groupe.

- Un étudiant ne pouvait jamais faire un examen dans une matière qu'il a réussie déjà.
- La révision des notes n'était pas possible quelque soient les circonstances.
- Résultats à partir de l'année 2005
 - Une composition subit une seule correction.
 - Les correcteurs peuvent diviser l'ensemble des compositions entre eux.
 - La note d'une composition est faite sur vingt.
 - La note finale d'une matière (demi-valeur ou 6 crédits) ayant subi un seul partiel est calculée selon la formule suivante :

$$NF = \text{SUP} \left(\frac{P + 2E}{3}, E \right)$$
 P : note de l'examen partiel
 E : note de l'examen final
 NF : note finale
- Les étudiants payent en L.L. chez Liban Post, la devise est conservée dans le nouveau système.

IV.2.3 Intégrer des nouvelles notions

L'intégration de la période de préinscription au cycle de l'inscription à beaucoup d'avantages, d'où cette période prépare les étudiants et l'institut à l'inscription. Donc, l'inscription est la période décisif de manière que les étudiants durant l'inscription sont obligés de s'inscrire, et l'administration est obligé à lancer le semestre universitaire, voilà le cycle du préinscription et d'inscription :

Cycle global [11]

- Au début de chaque semestre, les chefs des départements, les chefs des centres et le coordinateur des centres préparent le catalogue des matières, l'ouverture des sections, les affectations des professeurs aux sections pour chaque spécialité et dans chaque centre, ainsi que les horaires des études.
- Après la terminaison des horaires et les affectations, les nouveaux et anciens étudiants doivent s'inscrire aux matières disponibles durant le semestre courant.

- L'inscription s'étale sur une durée limitée dans le temps. Elle comporte une période de préinscription qui implique les choix des étudiants avant de payer les frais d'inscription. Cette période est utile pour avoir une idée des nombres des étudiants candidats aux matières.

- Le choix d'un étudiant pour ses matières dépend de l'horaire des études, des matières qu'il a déjà réussies, des conditions d'admission des matières et de son objectif principal visé.

La modification de l'un de ces éléments entraîne une modification du choix de l'étudiant durant la période d'inscription. Parfois, c'est l'horaire des études qui se modifie; parfois la période d'inscription commence avant que tous les résultats du semestre passé se terminent; parfois, il y a une modification dans la détermination des matières et de leurs prérequis et enfin l'étudiant commence à assister à un cours durant la période d'inscription, puis il change d'avis pour le substituer par un autre.

- La modification des choix des étudiants influence l'horaire ou plutôt les sections ouvertes: si une matière a un nombre petit d'étudiants, elle sera annulée de l'horaire d'études. Donc, la suppression entraîne à son tour des modifications des choix.

IV.2.4 Résoudre les anciennes fautes

J'ai déjà cité les fautes techniques de l'ancien SI dans la section [I.3 la problématique], mais beaucoup des fautes fonctionnelles qui résident dans l'ancien SI, et qu'il faut y trouver une solution, surtout l'incohérence entre certains modules, mais grâce à l'unification des données et des applications du SI attendu on peut les minimiser, voilà certains exemples d'incohérences [11]:

- Un étudiant à l'institut peut soit étudier à la carte comme auditeur libre, c'est-à-dire ne visant pas un diplôme et choisissant librement des UE, soit étudier afin d'obtenir un diplôme.

Il existe actuellement une règle de gestion qui impose que les conditions d'admission aux matières soient obligatoires. Parmi ces conditions, il y a des pré-requis, c'est-à-dire la nécessité d'avoir réussi en certaines matières avant d'effectuer le choix.

La question est comment distinguer formellement un auditeur libre d'un auditeur visant un diplôme pour pouvoir appliquer les règles concernées ?

- Dans le test d'admission, l'étudiant peut choisir d'être candidat à plusieurs spécialités au temps où les examens pour toutes les spécialités s'effectuent en même temps.

- Les périodes des études sont divisées en semestres, le catalogue des matières est semestriel, la répartition des professeurs est semestrielle tandis que les contrats des enseignants sont annuels.

Comment prévoir le second semestre avant de terminer le premier semestre et avoir les résultats qui sont une cause d'ouverture ou de non-ouverture d'une matière au second

semestre ?

- Les périodes des études étant semestrielles, pourquoi avoir des matières étalées sur deux semestres ou plutôt annuelles ?
- Une seule période d'inscription au début du semestre est ouverte, comment permettre l'inscription dans des matières plusieurs fois au même semestre (Ex. : Test BULAT, matières de langue) ?
- Etant donné que les matières possèdent des pré-requis, pourquoi définir les groupes des matières pour les étudiants de la première année en fonction du test d'admission ? Le système actuel définit un groupe de matières pour chaque cas possible du résultat du test.

IV.2.5 Problèmes rencontrés

Les problèmes sont rencontrés dans les différentes étapes de l'évolution de notre nouveau SI, surtout durant l'analyse, l'étape la plus essentielle pour élaborer les idées principales et le scope du nouveau SI, citons :

- Certaines centres d'enseignements étaient parfois complètement contre la centralisation des données par crainte de publier leurs données à l'extérieur du centre. Mais dans ce cas, ce n'était pas à eux à décider, la direction de l'institut à imposer toutes ces décisions, et ils ont été obligés de nous fournir toutes les informations nécessaires.
- Aussi d'autres centres d'enseignements refusaient d'utiliser ce système qui est supposé englobait toutes les données et devenir la seule référence informatique et académiques aux différents départements et centres d'enseignements de l'institut. Ils préféraient gérer le tout en interne, au sein du centre, et ne dévoiler aucune information à l'extérieur, même si cette dernière est de la direction centrale de l'ISAE-Cnam Liban. Mais ce problème a été réglé aussi par une décision définitive de la part conseil d'administration.
- Il a eu des cas où des centres et des chefs de départements ne comprenaient rien de ce qu'on leur proposait, et donc nous étions obligés de leur faire des formations intensives et des prototypes.

IV. 3 Conception

Maintenant et après les études préliminaires du nouveau SI, vient l'étape de la conception qui précède le codage. L'équipe du SI à l'institut a consacré beaucoup du temps pour la conception, à cause des complexités des procédures et des scénarios. Dans ce qui suit, je ne veux pas citer toutes les graphes des flux qui représentent les différents scénarios, mais seulement un exemple. Voilà les points dont ce chapitre voudra traiter :

- Architecture fonctionnelle du SI.
- Les services à réaliser.
- Graphe de flux de l'inscription.
- Cas d'utilisation de l'inscription.

IV.3.1 Architecture fonctionnelle du SI

IV.3.1.1 Diagramme de contexte (boîte noire)

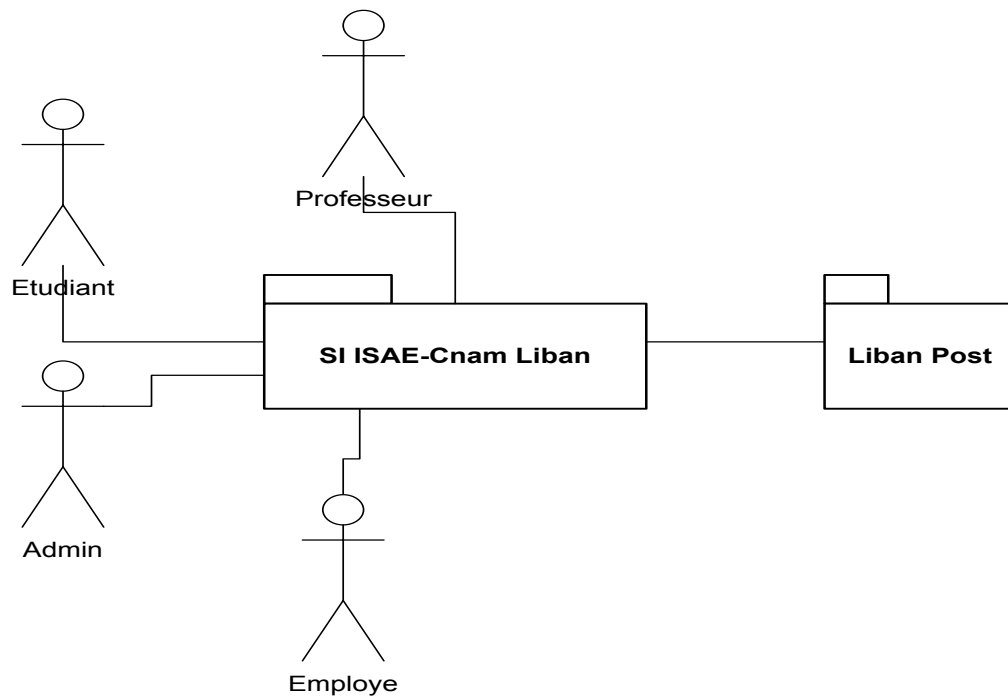


Figure 12: Diagramme de contexte (boîte noire)

IV.3.1.2 Diagramme de contexte avec sous système fonctionnels (boîte blanche)

Le diagramme présenté ci-dessus représente les différents modules du système envisagé, avec des relations entre les modules ou paquets, ces relations représentent l'échange des informations entre eux :

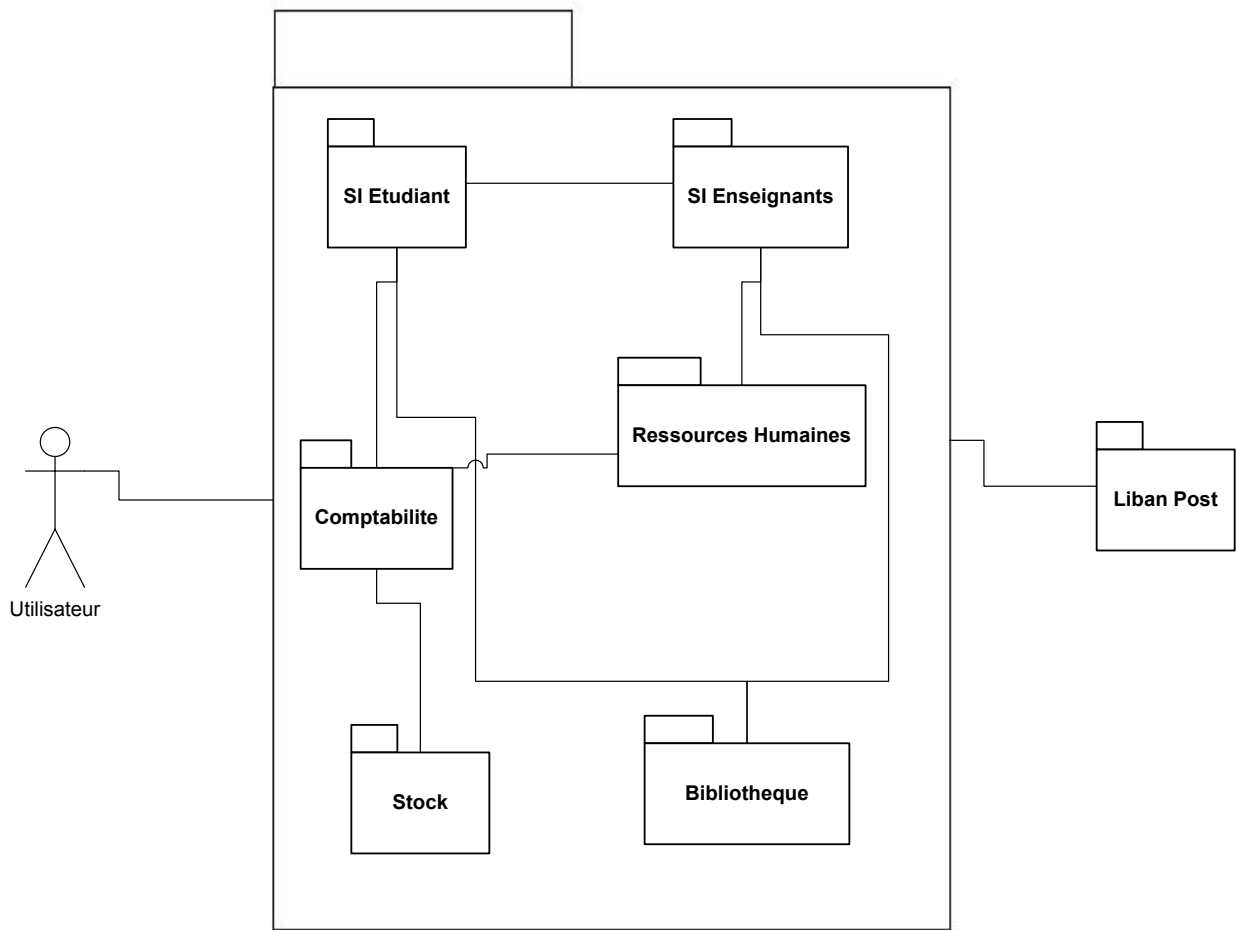


Figure 13: Diagramme de contexte (boîte blanche)

IV.3 .2 Les services à réaliser

- Inscription.
- Préinscription.
- Catalogue des diplômes.
- Equivalence des matières.
- Admission.
- Accord chef.
- Gestion d'horaire.
- Profile de l'étudiant.
- Résultat.
- Statistique sur les résultats.

- Email.
- Présence des professeurs.
- Salaire des professeurs.
- Stage pour les étudiants.
- Coordonnées des professeurs.
- Bibliothèque.
- Gestion du stock.

IV. 3 .3 Graphe de flux de l'inscription

L'inscription permet à l'étudiant de choisir parmi les UE ouvertes durant un semestre celles auxquelles il veut s'inscrire. La clôture de cette opération se fait quand la confirmation de paiement est émise du Liban Post à l'ISAE.

Description :

- L'étudiant peut visualiser la liste des UE ouvertes pendant le semestre, les UE auxquelles il lui est permis de s'inscrire, les UE qu'il a déjà réussies, les UE dont il a besoin de réussir les pré-requis, les UE dont il a besoin d'un accord du chef de département.
- L'étudiant choisit parmi les UE permises celles auxquelles il veut s'inscrire.
- L'étudiant valide la préinscription.
- L'étudiant reçoit la fiche de paiement du service de scolarité et paye la facture au Liban Post.
- L'institut reçoit les confirmations des paiements d'une manière périodique et électronique.

Notons que les acteurs impliqués dans le graphe suivant sont :

- L'étudiant
- Le LibanPost
- Le service de scolarité

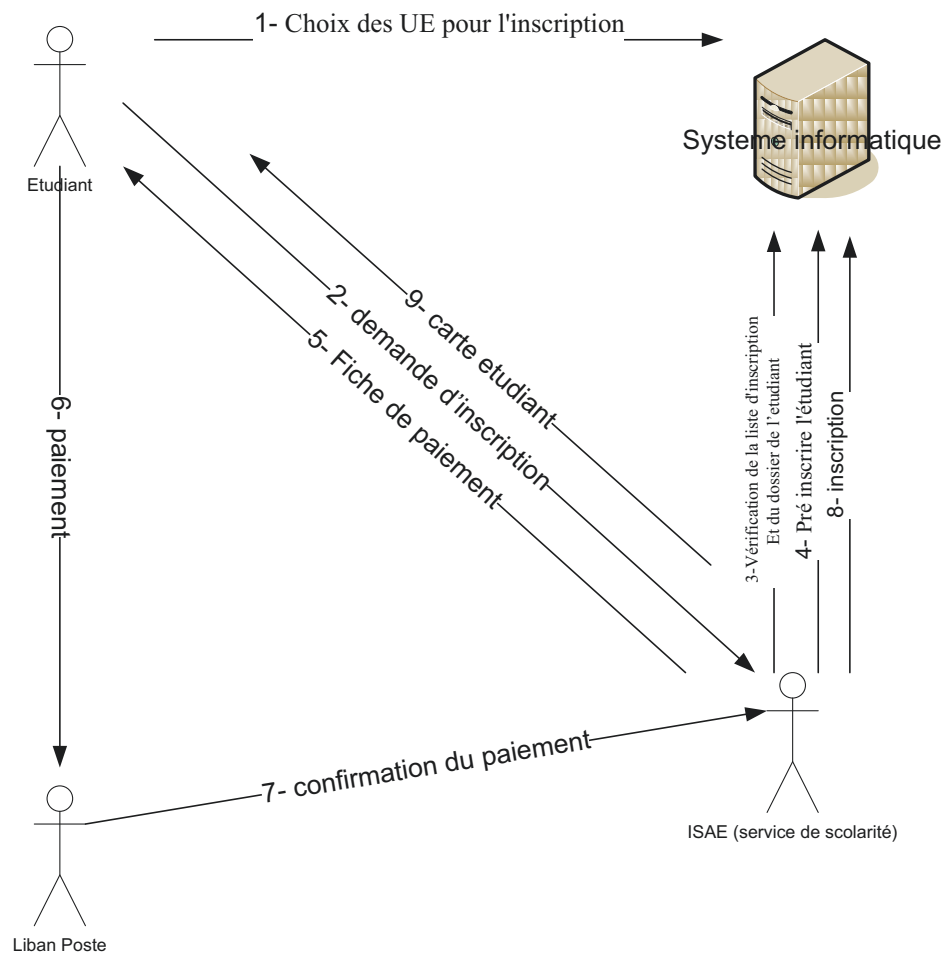


Figure 14: Diagramme de flux de l'inscription

IV.3.4 Cas d'utilisation de l'inscription

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans ce diagramme de cas d'utilisation, les utilisateurs sont l'étudiant et le service de scolarité, ils interagissent avec les cas d'utilisation suivants :

- Préinscription semestre
- Inscription stage
- Gestion de l'inscription à un semestre
- Gestion du fichier de l'étudiant
- Gestion de l'inscription à un stage

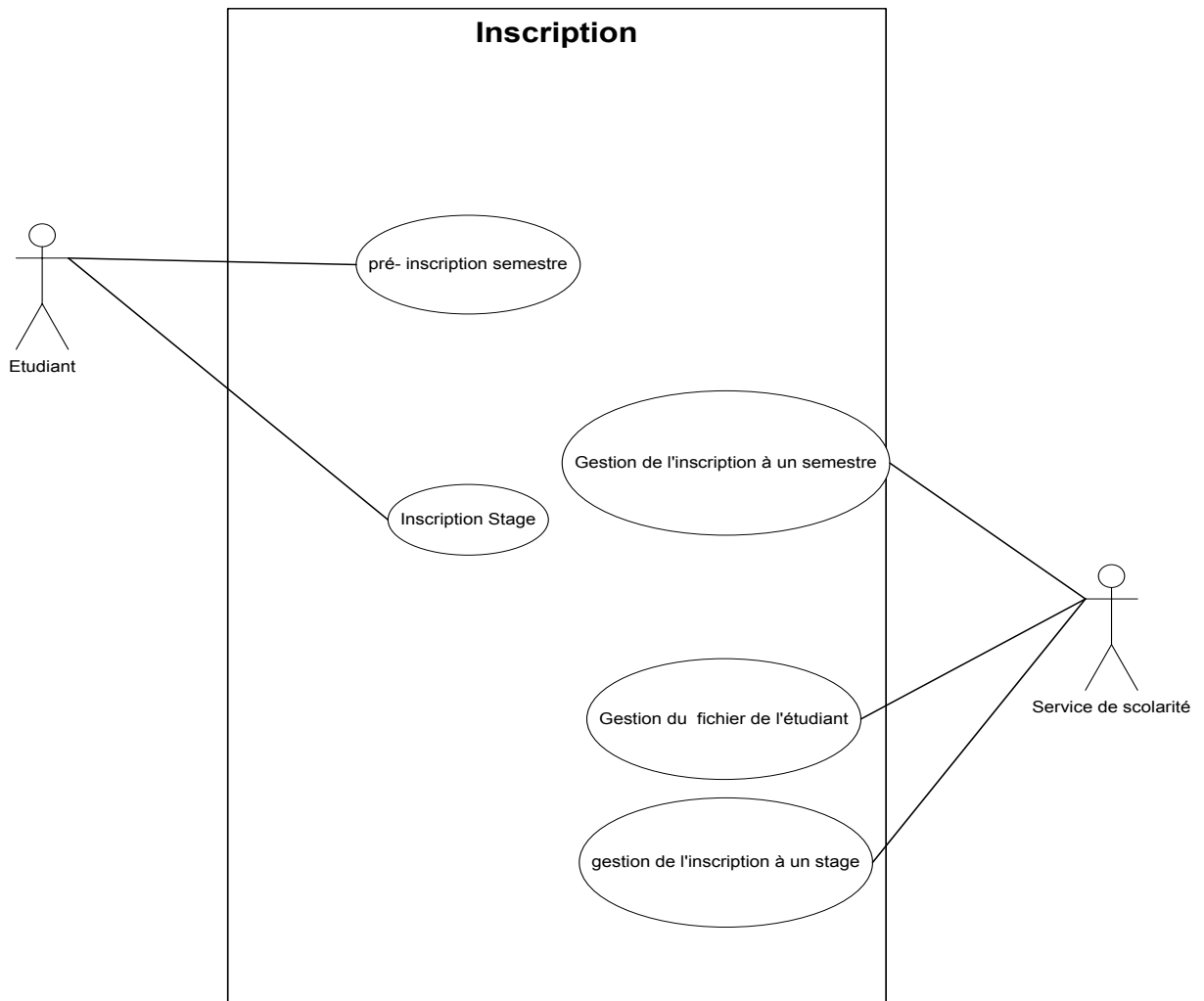


Figure 15: Cas d'utilisation de l'inscription

V Développement réalisé

V.1 Lancement du développement

Après la terminaison du cahier des charges en juillet 2008, l'équipe IT était près de commencer le développement. Le développement est lancé effectivement au début du premier semestre 2008. J'ai participé à l'analyse et la conception quand j'étais un étudiant à l'institut au cycle C, puis j'ai commencé le développement avec l'équipe en janvier 2009. Les premières tâches réalisées par l'équipe étaient la préparation des serveurs Web et SGBD, et puis le développement des couches hausses, voilà les points principaux dans le développement dont je voudrai traiter ultérieurement :

- Développement de la base de données
- Développement Web
- Développement des composants

V.2 Développement de la base de données

L'équipe SI a consacré beaucoup du temps pour développer la base de données, surtout le chef du développement qui est aussi l'administrateur de base de données, c'est une personne très qualifiée qui m'a aidé dans le développement. Cette partie va traiter les différents aspects du développement et de la configuration liés à la base de données, en n'entrant pas beaucoup dans les détails techniques.

V.2.1 Préparations des serveurs SGBD

Deux serveurs SGBD MySQL sont installés pour servir l'institut, le premier se situe au sein de l'institut à Beyrouth et le deuxième à l'Amérique. L'existence de deux serveurs aide à la sécurité des informations, c-à-dire si l'un tombe en panne on redirige nos applications sur l'autre. Les caractéristiques du serveur MySQL sont expliquées dans le chapitre II [Etat de l'art section II.2.3]. Ce serveur (MySQL) possède beaucoup d'avantages, il est performant et supporte beaucoup de charge, rappelons que NASA utilise aussi MySQL !

Installation sous Linux/Ubuntu et configuration

À l'aide de l'utilitaire Synaptic dans Ubuntu on peut chercher les paquets de MySQL puis les installer, cet utilitaire cherche automatiquement les paquets indispensables de MySQL et les coche pour l'installation. Après l'installation de MySQL on peut utiliser MySQL Query Browser pour exécuter des requêtes SQL et MySQL Administrator pour configurer MySQL (créer des utilisateurs et des bases de données, lancer le service, faire des restaurations et des backups etc.). Par ailleurs, la communauté de MySQL a lancé MySQL Workbench, ce produit qui va remplacer MySQL Administrator et MySQL Query Browser plus tard. MySQL Workbench ajoute des nouvelles fonctionnalités comme le designer de la base de données, nous savons

que c'est important de dresser la base de données visuellement pour mieux comprendre la structure des données et le modèle relationnel. Actuellement nous testons ce produit afin de l'utiliser après. Aussi, il existe d'autre logiciel léger pour manipuler MySQL comme PhpMyAdmin via le Web et MySQL Monitor via le console Linux, ce sont deux méthodes idéales pour se connecter à la base de données sans besoin d'un débit élevé comme MySQL Query Browser (l'interface graphique). Donc, j'ai profité de ces méthodes quand j'avais travaillé de la maison avec une connexion sans file très lente.

V.2.2 Création de la base de données

V.2.2.1 Modélisation

La modélisation des données décrit les objets du système d'information avec leurs caractéristiques, il s'agit donc d'une description statique de la réalité. Ces objets peuvent être reliés entre eux par des liens (associations). Le modèle entité-relation implémenté dans l'ancien base de données MS Access est migré vers MySQL, des changements sont réalisés dans le modèle pour que ce modèle statique soit plus proche de la réalité et mieux à comprendre et à manipuler, et ses entités sont plus significatives, comme la modification de la partie des catalogues des diplômes et la partie des matières et de spécialités. Ultérieurement, nous allons expliquer l'évolution de la base de données. Donc le modèle des données de l'ISAE est toujours évolutif, pour qu'il puisse être adaptable aux besoins. Par ailleurs, ce modèle a subi beaucoup de modifications et d'altérations. La base de données principale est nommée « db_isae », au fur et à mesure nous avons créé deux autres bases de données : « Catalogue_diplome » et « resu ». En effet, avoir plusieurs bases de données a beaucoup d'intérêt, comme la modularité du SI, et la facilitation de la maintenance et la compréhension du modèle.

V.2.2.2 Moteurs de stockage

On a deux moteurs de stockage utilisés à l'institut : InnoDB (par default) et MyISAM. Voilà les avantages et les inconvénients de chaque moteur : [9]

1. MyISAM

Les avantages :

- Très rapide pour les requêtes de type SELECT ou INSERT.
- Il supporte les index Fulltext : permet d'effectuer des recherches sur des mots en se basant sur un index spécifique, accélérant ainsi les recherches.
- Il gère les conflits d'accès : permet de verrouiller une table pour des opérations bien précises.

Les inconvénients :

- Il ne supporte pas les transactions.

- Il ne supporte pas les clés étrangères.

2. InnoDB

Les avantages :

- Il supporte ACID : permet d'assurer que chaque enregistrement sera complètement réussi, ou complètement échoué, ainsi les risques d'erreurs sont impossibles, même en cas de panne.
- Il gère les transactions (instructions SQL BEGIN, COMMIT, ROLLBACK etc.).
- Il supporte les clés étrangères et les intégrités référentielles.
- Il possède un système de récupération automatique en cas de crash.

Les inconvénients :

- Il ne permet pas les index fulltext.
- Son administration est un peu plus complexe (gestion de tablespace, paramètres supplémentaires dans le my.cnf etc.)
- Le moteur de stockage est plus lent que d'autres et gourmand en ressources mémoires et en espace disque.

En conclusion, les deux moteurs cités sont utilisés dans nos bases de données, MyISAM pour les tables comme « LibanPost » qui ne possède pas des clés étrangers et ne supporte pas de transactions, tandis InnoDB est le moteur « officiel » de nos tables par défaut, car nos applications fonctionnent avec des transactions et des clés étrangères.

V.2.2.3 Créations des tables et des relations

La création des tables et des relations était une opération complexe, surtout s'il y a beaucoup des relations entre certaines tables (Inscription, Etudiants etc.) . Ces tâches, sont de la responsabilité de l'administrateur base de données, et avec notre support. La « db_isae » est la base principale qui est constituée à peut près de 80 tables qui représentent le modèle statique de l'institut. Notons aussi que le standard UTF-8(8-bit Unicode Transformation Format) est choisi comme standard de codage pour les bases de données, aussi au niveau des pages JSP, ce type de codage est déterminé. Par ailleurs, on peut définir par défaut le codage par l'interface d'administration du MySQL (MySQL Administrator maintenant MySQL WorkBench), en allant au menu des variables de démarrage. Aussi on peut déterminer le type de codage ou CharSet pour chaque table séparément.

Rappelons aussi que l'indexage des tables est important pour accélérer les recherches des données, d'où on ne peut pas répéter le même indexe dans une table, par exemple dans la table « users » on a le champ « username » comme indexe, voilà la table « users » avec ses détails :

Les attributs de la table «users» :

- UserId : BigInt, clé primaire.
- Username: VarChar.
- Password :VarChar.

- Function :VarChar.
- No : Integer, clé étrangère.
- Fermer : TinyInt.

Moteur du stockage : InnoDB.

Codage : UTF-8.

La figure ci-dessous représente les tables qui gèrent les données des épreuves (examen) de la base de données « resu » :

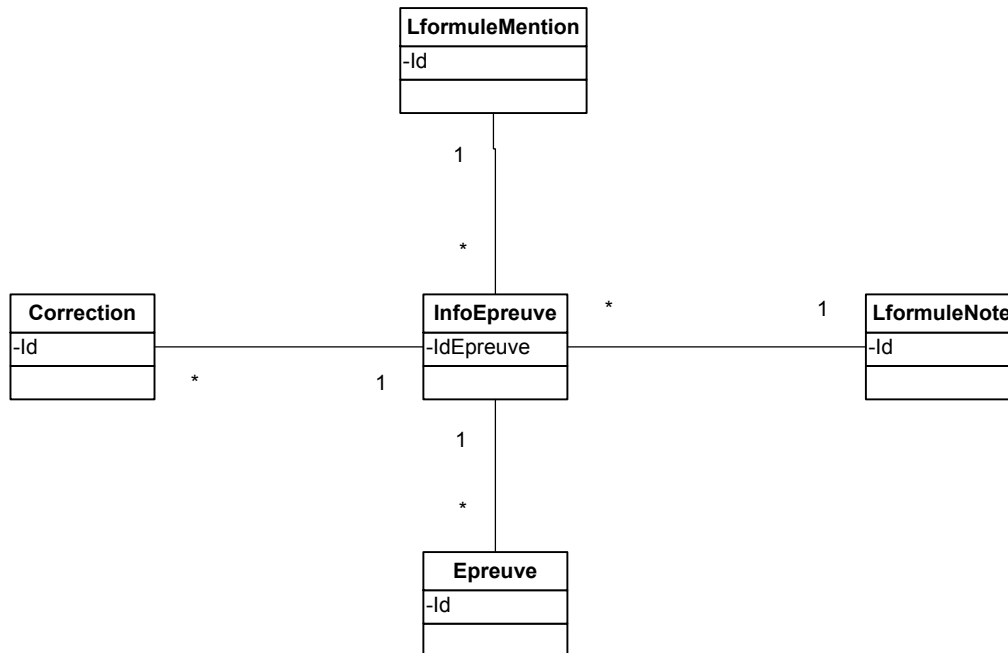


Figure 16: Modélisation des épreuves

V.2.3 Contrôle d'accès aux données

L'information est l'unité centrale du SI, comme nous pouvons la nommer, sa protection est de notre responsabilité. L'accès aux données se fait de deux manières principales :

- Direct : Accès direct via QueryBrowser, MySQLAdministrator, MySQLDUMP, ce type d'accès est privilégié pour nous l'équipe du SI, pour effectuer les transactions directes (CRUDE) et les opérations de configurations, par exemple pour les tâches qui ne possèdent pas des interfaces graphiques, comme dans le cas du transfert d'un étudiant d'un centre à un autre, il suffit de changer le numéro de centre du table « étudiant».

- Indirect : Accès indirect via les applications Web développées d'où cet accès est contrôlé par les tables « users » et « droitaccès ». La table « users » représente les utilisateurs qui ont accès à la base de données avec une fonction, les fonctions valables sont :
 - Chef : pour les chefs de départements.
 - Etudiant : pour les étudiants.
 - Employé : pour les employés de l'institut.
 - Professeur : pour les professeurs.

La table « droitaccès » joue le rôle dans le cas où l'utilisateur est un employé ou chef ou professeur, elle détermine le droit d'accéder aux applications par utilisateur, le champ « type droit » énumère les droits possible, citons à titre d'exemple les droits suivants :

TypeDroit
chefCentre
chefscolarite
Diplôme
examenGuest
assistant
scolarite
catalogue
examenBarcode
Prof
examenScolarite
Admin
chefpresence
examenResultat
resuDispense
resultats
Horaire
presence
dispense
examenAdjoint
examenRevision
presenceGuest

Table 5: Type Droit

Interprétation :

L'utilisateur qui possède le droit « admin » peut accéder à toutes les applications Web,

l'utilisateur du droit «resultat» peut accéder seulement aux applications concernant les résultats. Cette méthode comme elle possède des avantages, elle possède aussi des inconvénients. L'inconvénient principale, est que l'augmentation des applications et des fonctionnalités entraînent à une augmentation des types des droits, donc pour chaque utilisateur on a plusieurs tuples qui représentent les nombres des applications ou des fonctionnalités permises, mais c'est compliqué. Actuellement nous sommes en train d'étudier la notion des groupes d'utilisateurs, c'est-à-dire au lieu de lier les utilisateurs directement aux applications, nous les connectons les connectent à des groupes d'applications.

V.2.4 Bases de données distribuées

Travailler avec des bases de données moyennes et compréhensibles étaient plus faciles qu'une seule base, qui englobe toutes les fonctionnalités. Au début, il y avait seulement la « db_isae » avec 80 tables. Afin de faciliter la manipulation des données nous avons créés deux autres bases de données sont la «resu » pour les résultats des étudiants et la base «catalogueDiplome » pour les diplômes et leurs catalogues. Par ailleurs, pour éviter la duplication des certaines tables, nous avons utilisé la notion «Federated Table » du MySQL.

Tous d'abord le « Federated Table » est un moteur de stockage, d'où la table du le moteur est de type « Federated ». Cette table est comme une interface liée à la table source de la base source. La requête ne s'exécute pas réellement sur le «Federated Table » mais sur la table source, où elle est déterminée dans le code du table « Federated ». Par exemple, la table « Matiere » dans la base de données « resu » est simplement connectés à la table « Matiere » de la base centrale, dont elle extrait les données. Donc d'après cette méthode, nous avons résolu le problème de la duplication des informations. Donc la décomposition de la base principale a aboutit à un ensemble des modules faciles à manipuler, par suite nous pouvons développer des applications Web reliées à une base de taille moyenne et petite.

V.3 Développement Web

V.3.1 Introduction

Cette partie va traiter les différents aspects du développement Web à l'institut, et ma contribution dans le projet. Je ne veux pas donner ici des leçons de développement, mais prouver que la phase de développement est l'étape où toutes les idées sont concrétisées sur le plan de la programmation et d'architecture, aussi en jetant un coup d'œil sur le code et les différents langages de programmation utilisés et les nouvelles technologies Web intégrées comme AJAX. Enfin, je voudrai parler des problèmes rencontrés et les améliorations possibles.

V.3.2 Architectures

Rappelons, que le langage JAVA est un langage architectural contrairement au concurrent PHP, c'est-à-dire pour en profiter, il est meilleur de développer suivant une architecture comme j'avais procédé dans nos applications Web, afin de diviser le développement en couche et de séparer le code pour obtenir une application lisible, et facile à développer et à maintenir au futur. Tandis que le chef de développement et l'autre développeur avaient choisi le modèle 2-tiers, d'où le code qui contrôle le comportement de l'utilisateur se trouve dans la page JSP, cette méthode a davantage d'accélérer la programmation, mais le problème consiste que la page programmée sera complexe et difficile à maintenir et à comprendre sémantiquement au futur.

Donc, l'architecture que j'avais adopté est MVC, l'avantage du MVC et la séparation entre les couches de présentation, contrôle et du modèle :

- La couche présentation est modélisée par des pages JSP.
- La couche contrôle est modélisée par des « Servlets » et des Filtres leurs rôles est : gérer les requêtes provenant du client. [Annexe 1]
- La couche modèle est modélisée par les « Beans » qui représentent les modèles de données.

Au début de développement suivant le modèle MVC, j'ai rencontré beaucoup de difficultés, car il faut bien analyser avant de commencer le développement, d'où l'augmentation du temps d'analyse, mais le profit est après, surtout quand nous obtiendrons un logiciel lisible et modulaire.

En effet, mon but est de créer des modules, des petites applications au futur ou des composants, je voudrai expliquer ultérieurement le développement des composants.

Les deux figures suivantes représentent les deux architectures adoptées au département SI, moi je préfère la première architecture:

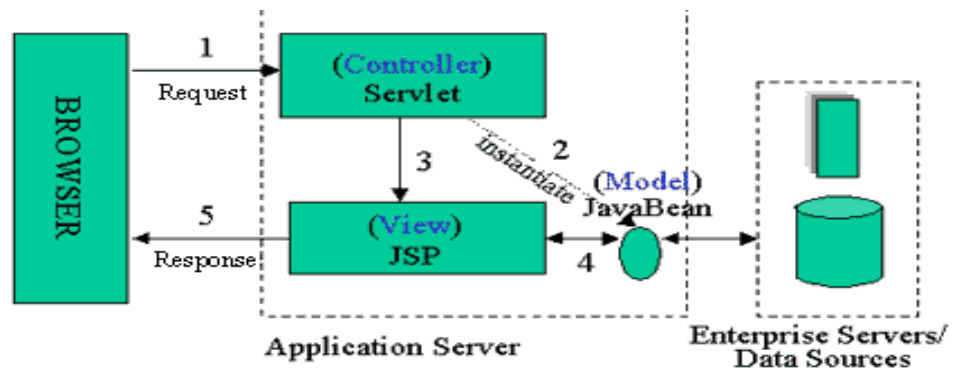


Figure 17: Architecture JSP/MVC

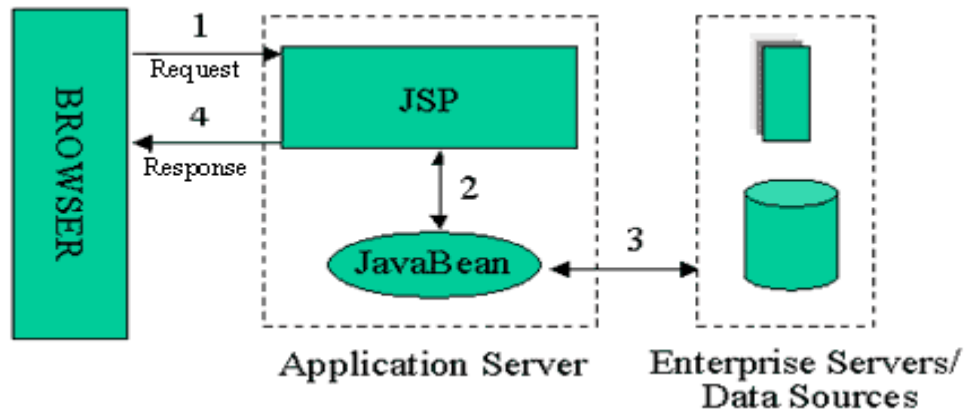


Figure 18: Architecture JSP sans séparation des couches

V.3.3 Les bibliothèques

Beaucoup des méthodes Java sont utilisées dans plusieurs projets Web, or c'est une perte du temps de réécrire les mêmes méthodes, par exemple la méthode qui retourne la liste des centres «AvoirCentre() ». Donc, les bibliothèques contiennent les méthodes communes aux différentes applications. Nous avons deux bibliothèques principales :

- « SIIsaeLib » développée par moi, cette bibliothèque engendre des méthodes de base utiles pour plusieurs projets, comme avoir des listes des objets ou des tables de la base de données, et des méthodes mathématiques et algorithmiques, par exemple la méthode HashToList() qui prend en paramètre un Hashtable (tableau clé/valeur) et retourne une liste des objets, cette méthode est essentielle pour les calculs des résultats des étudiants; où chaque étudiant est présenté par un objet dans la liste retourné par cette méthode.[Annexe 2]
- « isaelibrary » développée par le chef du développement et l'autre développeur; cette bibliothèque comporte beaucoup de fonctionnalités utilisées dans la plupart des projets. Par exemple les méthodes qui gèrent les droits d'accès des utilisateurs, de préinscription, liste des diplômes etc.

V.3.4 Utilisation d'AJAX

V.3.4.1 Définition

AJAX (Asynchronous Javascript and Xml) est un terme inventé par Jesse James Garret en février 2005 pour désigner une technologie ancienne de 1999. Il s'agit d'utiliser le code JavaScript sur le client pour invoquer le serveur en tâche de fond et obtenir des données au format XML afin de modifier le Dynamic HTML et rafraîchir une partie de la page.[10]

Cela consiste à exploiter l'objet XMLHttpRequest, présent dans les navigateurs récents. Ce composant permet de jouer le rôle d'un client HTTP, d'envoyer une requête et d'attendre la réponse pour pouvoir la manipuler.

AJAX permet d'optimiser le trafic en ne ramenant que les données nécessaires au rafraîchissement d'une page et non une page entière. De 20 à 80% d'une page sert à la présentation. AJAX permet d'optimiser les affichages de type "liste/détail", de trier les données sur le client, d'aider l'utilisateur lors des saisies, d'enrichir les arbres ou d'alimenter les onglets (une liste horizontale de liens)... AJAX permet également d'offrir une approche "une seule page pour l'application". Pour, par exemple, proposer un tableur ou un traitement de texte dans une page WEB.

Sous le terme AJAX, on intègre également d'autres utilisations du JavaScript. Il s'agit d'améliorer l'ergonomie de l'application en ajoutant des composants graphiques complémentaires (aide à la saisie, saisie de date, arbre, onglet, menus, piles, accordéon, curseur, barre de progression, etc.)

Si toutes les données sont disponibles, les technologies Dynamic HTML (DHTML) et CSS permettent une mise jour dynamique de la structure de la page. Il est possible de proposer une ergonomie proche de celle d'une application cliente classique. Des parties de pages sont cachées ou affichées à la demande ; des fenêtres apparaissent ou disparaissent sur événements ; des zones sont complétées dynamiquement à l'insu de l'utilisateur, des arbres s'enrichissent au fur et à mesure, etc. Si les données ne sont pas disponibles, une requête AJAX asynchrone y remédiera. Cette technologie a pour but d'améliorer l'efficacité des internautes et l'attractivité des sites. En réduisant de quelques secondes les manipulations des utilisateurs, un centre d'appel améliore sa rentabilité. En attirant de nouveaux internautes, un site améliore son ROI. Les applications traditionnelles offrent une très grande richesse de l'interface. Des logiciels comme Open Office proposent des fonctionnalités que l'on ne retrouve pas sur les sites internet. À l'opposé, les applications HTML offrent une très grande facilité d'accès. AJAX est une technologie permettant de rapprocher ces deux objectifs (figure 19).

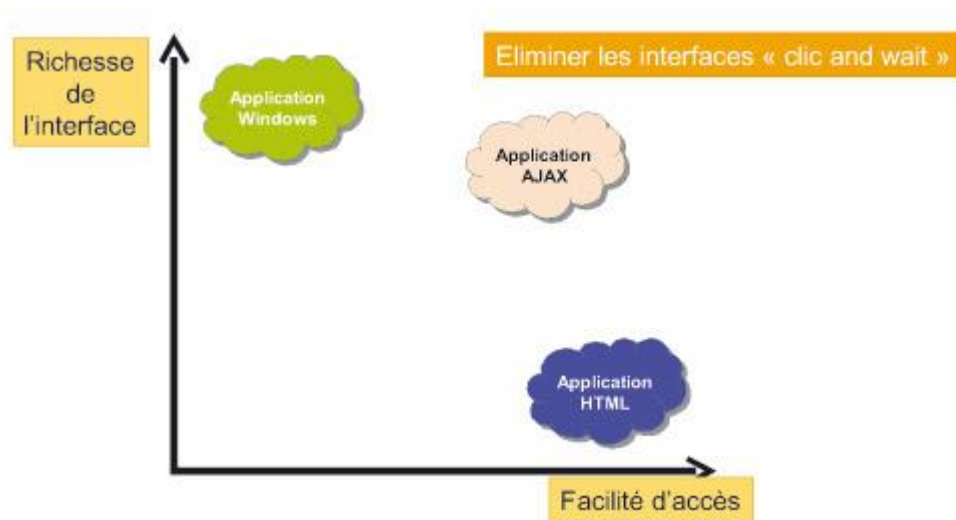


Figure 19: Positionnement d'AJAX

V.3.4.2 Implémentation au SI de l'ISAE-Cnam

Nous avons implémenté AJAX dans les applications Web suivantes :

- Resultat : pour avoir les résultats des examens.
- ResultatEtd : pour obtenir les résultats par chaque étudiant.
- HoraireProf : pour obtenir l'horaire d'enseignement de chaque professeur à l'institut.
- Cataloguediplomes /Cataloguediplomespublic : c'est une application responsable de gérer les diplômes et leurs matières, cette application possède deux versions une privée et l'autre public.

Annee	Centre	Semestre	Session	Cour Intensif	Code Cnam
2010	Beyrouth	2	1	0	rsx102
Voir Notes					
RSX102 Technologies pour les applications client-serveur					
NoDossier	Nom	Prénom	Note Finale	Réussi	
7857 f	EL GHARIB	Rony	10	Oui	
7892 f	ABDALLAH	Tarek	Abs.	Non	
5212 f	ABDALLAH	Mouhamad	11.5	Oui	
6632 f	AL HAJJ	Hassan	Abs.	Non	
468 S	AMACHA	Maysaa	Abs.	Non	
483 t	ANTOUN	Joseph	10	Oui	
388 BAAL	AWADA	Fatima	Abs.	Non	
6183 f	BARDAWIL	Bernard	11	Oui	
7602 f	BOU ZEID	Fadi	12	Oui	
209 BAAL	CHAIB	Malak	2.5	Non	
4723 f	EL KAREH	Sahar	Abs.	Non	
4717 f	EL-BADAWI	Elias	Abs.	Non	
7863 f	FAKIH	Ali	7	Non	
4276 F	FAWAZ	Ahmad	Abs.	Non	
7882 f	GEBRAN	Fadi	11	Oui	

Figure 20: L'application « Resultat » intégrant AJAX

V.3.5 Déploiement

V.3.5 .1 Choix du serveur d'application

Les deux serveurs d'applications Apache Tomcat et GlassFish, sont les plus populaires pour les applications JAVA et qui cohabitent JSP, mais après certains tests réalisés par moi-même, j'ai remarqué que Tomcat était plus rapide et léger que GlassFish, qui était lourd et exécute beaucoup des «choses» lors du lancement ! Mais enfin, ce n'était pas mon choix de choisir Tomcat comme serveur d'application, c'est le chef du projet qui l'a choisi, et je pense que c'est le meilleur choix !

V.3.5 .2 Importance du déploiement

Comme développeur, je pense que le déploiement est une phase très importante, car elle nous donne l'idée si notre application peut être cohabitée chez le serveur d'application avec ses différents composants (pages JSP, Servlets etc.) ou pas. À l'ISAE-Cnam Liban, nous avons deux types de déploiements : le déploiement local sur notre machine, et le déploiement sur le serveur public, qui est le plus critique, ici l'erreur est interdite ! Car nous travaillons en ligne. En conclusion, comme l'analyse et le développement nécessitent du temps, aussi le déploiement, et si chaque étape est réalisée bien et dans sa marge de temps, alors nous pourrons délivrer un produit dans le temps excepté.

La figure suivante représente la fenêtre d'administration des applications déployées chez Tomcat, d'où on peut soit arrêter, soit redémarrer, soit supprimer, soit recharger une application déjà déployée. On peut aussi visualiser le nombre de sessions ouvertes pour chaque application, les adresses IP qui demandent nos applications etc.

Tomcat Web Application Manager

Message:

OK

Manager

List Applications

HTML Manager Help

Manager Help

Server

Applications

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/Administrateur	Administrateur	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/AdmissionDossier	AdmissionDossier	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/HChef		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Figure 21: Serveur Tomcat : fenêtre d'administration des applications Web

V.4 Développement des composants

V.4.1 Du concept au développement

Le concept des composants est déjà expliqué dans la section «III.4 », dans cette partie je veux parler de ma participation dans le développement des composants et son intérêt.

Tout d'abord, avant de programmer, il faut bien analyser l'aspect fonctionnel d'un projet pour aboutir à un découpage modulaire. Concernant l'inscription, l'application de l'inscription est constituée de deux modules: l'inscription et le sous module Fiche. Quand un étudiant veut obtenir son fiche, l'employé appelle la fiche d'inscription en utilisant l'application d'inscription, mais effectivement, nous avons deux applications Web connectées via la balise import du JSP, d'où l'application principale « l'inscription » appelle celle du «Fiche » en passant le paramètre du numéro de la fiche, voilà la balise d'import qui représente la connexion entre deux composants Web:

```
<c:import url="http://localhost:8080/Fiche?nofiche=" + nofiche + "&msg=" + msg;"/>
```

La figure suivante représente les deux applications d'inscription et la connexion entre eux :

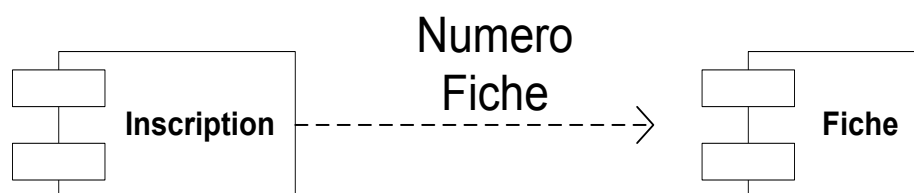


Figure 22: Schéma de composants d'inscription.

Aussi, on peut rediriger la requête HTTP d'un projet à un autre en utilisant « `httpresponse.sendRedirect(url_destinataire)` » dans un filtre Java. Je voudrai détailler le filtre après, dans le chapitre de sécurité. Le filtre est simplement c'est un « Servlet » ou une classe JAVA qui est exécuté au premier temps, avant l'appel des pages, son rôle essentiel est de filtrer les requêtes et les rediriger vers différents destinataires, par exemple si la source ne passe pas le nom d'utilisateur et le mot de passe en paramètre dans la requête source, le filtre la redirige vers une page login.

V.4.2 Avantage pratique

Nous avons beaucoup mentionné que la programmation par composants possèdent des avantages etc., or j'ai expérimenté ces avantages au cour de développement. Dans le cas de chaque mise à jour d'un projet Web classique, nous supprimons tout d'abord l'application entièrement du serveur d'application, cette application est déployé et sera inaccessible au client, puis nous effectuons la mise à jour et nous déployons de nouveau l'application, cette

méthode a un inconvénient d'arrêter toute l'application, tandis que si nous avons deux composants il suffit de faire la mise à jour sur le composant convenable, sans toucher l'autre composant. Par exemple, le chef de développement m'a demandé de faire certaine amélioration du projet Fiche, j'ai simplement arrêté le composant Fiche, effectué la mise à jour et la redéployé de nouveau sans arrêter l'autre application d'inscription.

V.4.3 Architectures composants implémentées à l'institut

Les deux figures suivantes représentent deux architectures utilisées dans le SI l'institut, la première est la plus utilisé car elle est plus sécurisée (pas de cookies), mais une des inconvénients est la répétition du même code de login dans chaque projet, et ce n'est pas vraiment un modèle composant. Tandis que la deuxième figure présente un modèle composant d'où on a des échanges des paramètres entre eux, mais l'inconvénient ici et qu'on a des cookies qui ne sont pas sécurisées, cette deuxième architecture n'est pas encore mature, elle a besoin encore d'études au niveau de sécurité.

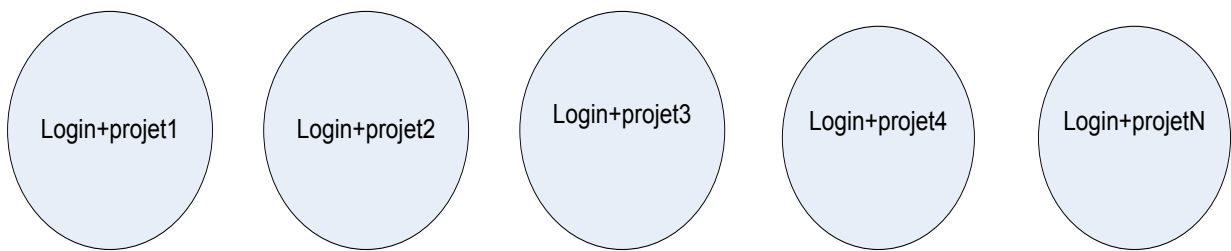


Figure 23:Projets indépendants avec répétition de codes

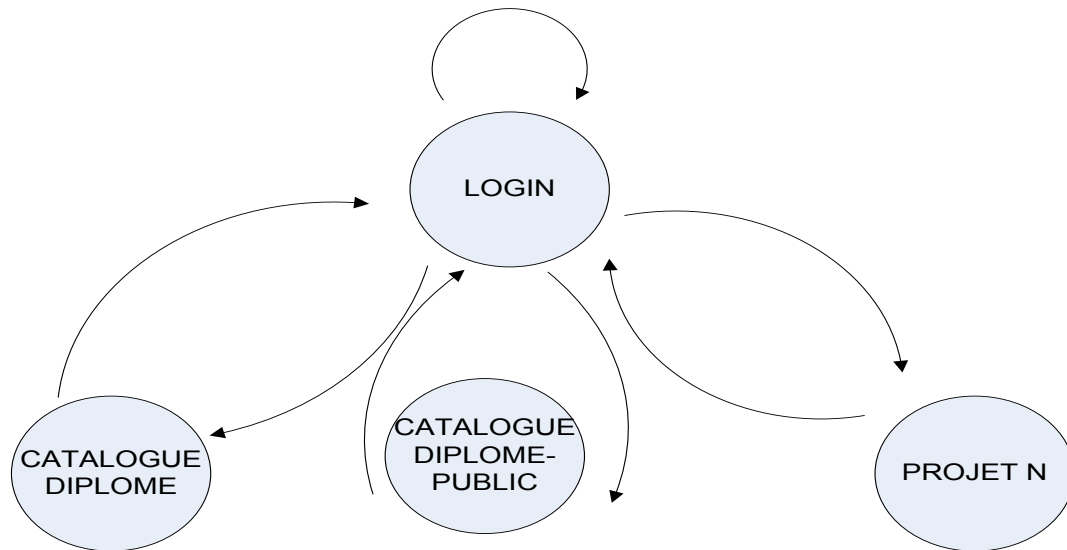


Figure 24: Modèle composant avec échanges d'informations

V.4.4 Source d'informations des composants

Chaque application Web nécessite une connexion à la base de données, et chaque composant rassemble à une application Web, or il faut faire bien attention aux sources des données des composants, je préfère d'après mon expérience que ses sources soient similaires, car nous pourrions tomber dans des erreurs si les bases de données ne sont pas cohérentes. Je veux donner un exemple : durant le développement des composants, je réalise toujours le test sur la base « db_isaeTest », et avant de déployer les composants en lignes, je change les connexions des composants vers la base principale «db_isae », mais si une fois j'oublie de rendre la connexion d'une application vers «db_isae », je tombe dans l'erreur qu'on a deux composants avec deux sources d'informations différentes!. Par exemple, le cas où la connexion de l'application d'inscription est différente que l'application « Fiche », dans ce cas la fiche ne pourra pas être trouvée !. Les deux figures suivantes représentent les deux cas des composants avec deux sources de données (cas non préférable), ou une seule source de données :

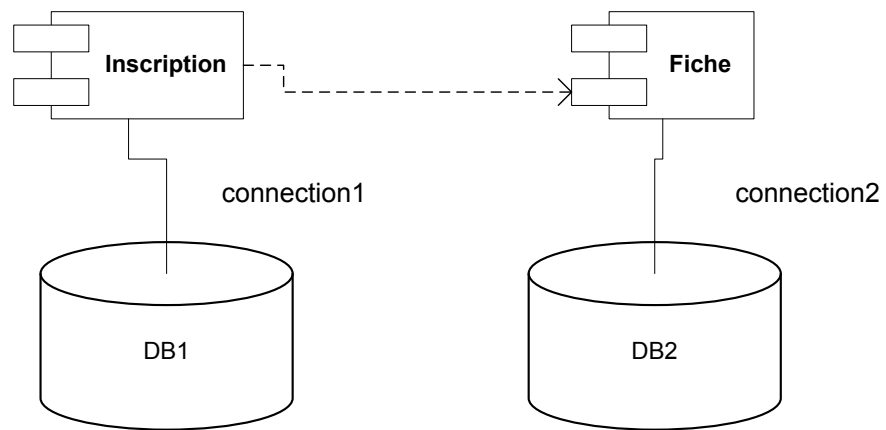


Figure 25: Modèle composants avec deux sources d'informations

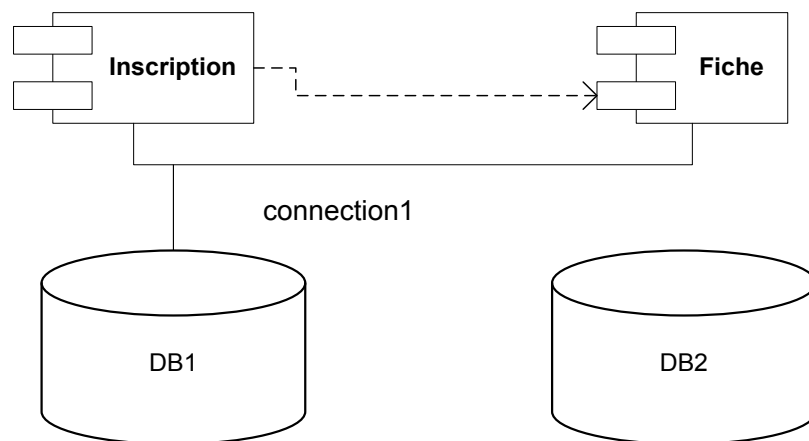


Figure 26: Modèle composants avec une seule source d'informations

VI Sécurité

VI.1 Introduction

Les attaques informatiques constituent aujourd'hui l'un des fléaux de notre civilisation moderne. Il ne se passe plus une semaine sans que l'on apprenne que telle entreprise ou tel institut a essuyé de lourdes pertes financières en raison d'une déficience de la sécurité de son système d'information.

La sécurité informatique vise à se protéger contre les risques liés à l'informatique, pouvant être fonction de plusieurs éléments tels que les menaces qui pèsent sur les actifs à protéger, les vulnérabilités de ces actifs, la sensibilité de ceux-ci.

Débuts de sécurité de réseau d'authentifier tout utilisateur. Une fois qu'authentifier, le mur à l'épreuve du feu impose des politiques d'accès telles que quels services sont permis d'être consultés par les utilisateurs de réseau. Cependant efficace pour empêcher l'accès non autorisé, ce composant ne vérifie pas le contenu potentiellement nocif tel que des vers d'ordinateur étant transmis au-dessus du réseau. Dans cette partie nous allons traiter la sécurité des applications Web et la sécurité des communications.

VI.2 Sécurité des applications Web

Une application est constituée des éléments principales suivantes :

- Pages JSP
- Classe JAVA

Le scénario d'appelle d'une page JSP commence par l'authentification, en validant une forme « login », et en passant un nom d'utilisateur et un mot de passe, voir section [V-2.3 Contrôle d'accès aux données]. Par ailleurs, pour contrôler la session (Session Tracking en anglais), qui engendre des données validées par l'utilisateur ou validées par le système comme la session ID, le mot d'utilisateur et le mot de passe, nous avons intégré les filtres Java.

Le filtre [Figure 27] est une classe JAVA, il est lancé automatiquement au début de chaque appelle, lors du lancement de l'application Web. Son rôle essentiel est de filtrer les requêtes sources suivant des règles spécifiques, le filtre peut refuser ou accepter les requêtes en basant sur les données de la session. Aussi nous pouvons implémenter plusieurs filtres dans une application si nous voudrions appliquer une sécurité en profondeur. Le filtre peut aussi examiner l'IP source, et d'autre paramètres du requête HTTP source, nous pouvons aussi développer des algorithmes de sécurité. Donc cette entité est un élément essentiel dans notre stratégie de sécurité, mais la limitation consiste que la session ne permet pas d'échanger les paramètres entre les différentes applications Web. Les « cookies » proposent une solution pour échanger ce type de donnée, mais l'utilisation des « cookies » reste limités, qui ne sont pas assez sécurisées, les attaquants peuvent « voler » les fichiers cookies facilement. L'étude d'une

solution final n'est pas encore terminée, concernant le passage des données entre les différents applications d'une manière sécurisée.[Annexe 3]

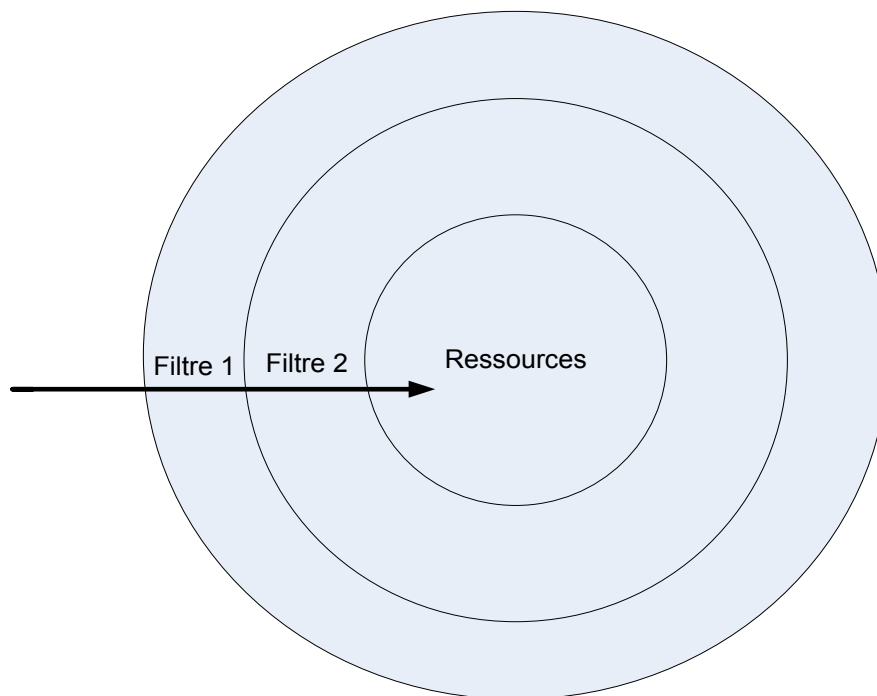


Figure 27: Sécurité par les filtres

VI.3 La sécurité des communications

VI.3.1 SSH

La sécurisation de nos communications est essentielle, puisque nous avons besoin de l'accès distant, et surtout que notre serveur en ligne se situe à l'Amérique. L'accès distant nécessite des méthodes de cryptage/décryptage, le protocole SSH propose une meilleure sécurité.

SSH est un protocole de communication qui a été conçu comme alternative sécurisée aux programmes Telnet/Ftp.

- SSH désigne à la fois:
 - Le protocole de communication sécurisée
 - Le service SSH sur le système à administrer
 - Le client utilisé pour l'administration du système distant
- Les objectifs de SSH sont:
 - Sécuriser les sessions d'administration à distance de systèmes à travers :
 - Le cryptage des mots de passes utilisés
 - Le cryptage des commandes échangées
 - Sécuriser les sessions de transfert de fichiers

- Sécuriser tout type de trafic applicatif à travers l'établissement de tunnels sécurisés pour véhiculer ce trafic.

VII Les fonctionnalités principales du produit

VII.1 Introduction

Les aspects techniques du logiciel sont déjà expliqués au cours du développement dans le chapitre V [Développement réalisés], mais il faut aussi expliquer les fonctionnalités de notre produit.

Donc cette partie va traiter les aspects fonctionnelles et modulaire du SI de l'institut. Ce SI est composé de plusieurs modules et de sous-modules, repartitionnés en groupes suivant le type d'utilisateur. Par exemple, l'utilisateur «ÉTUDIANT» possède de divers fonctionnalités. Ultérieurement, nous voudrions discuter les différents utilisateurs et les services fournis à eux, en donnant une vue globale des modules, puis en détaillant les modules les plus essentiels du SI. Rappelons dans ce contexte, que l'ergonomie était prise en considération dans la plupart de nos applications, comme l'inscription, gestion des horaires, affectation des matières.

VII.2 Description globale

VII.2.1 Les Modules

Les modules principaux du système de l'ISAE-Cnam Liban sont :

- Système d'Information des Etudiants
- Système d'Information des Enseignants
- Ressources humaines
- Comptabilité
- Stock
- Bibliothèque

VII.2.2 Première version

La première version était délivrée durant l'année universitaire 2008-2009, elle couvrait une partie du système d'information des étudiants. Cette version est schématisée par le graphe si dessous, les parties en couleur grise ne sont pas encore terminées à la première version.[14]

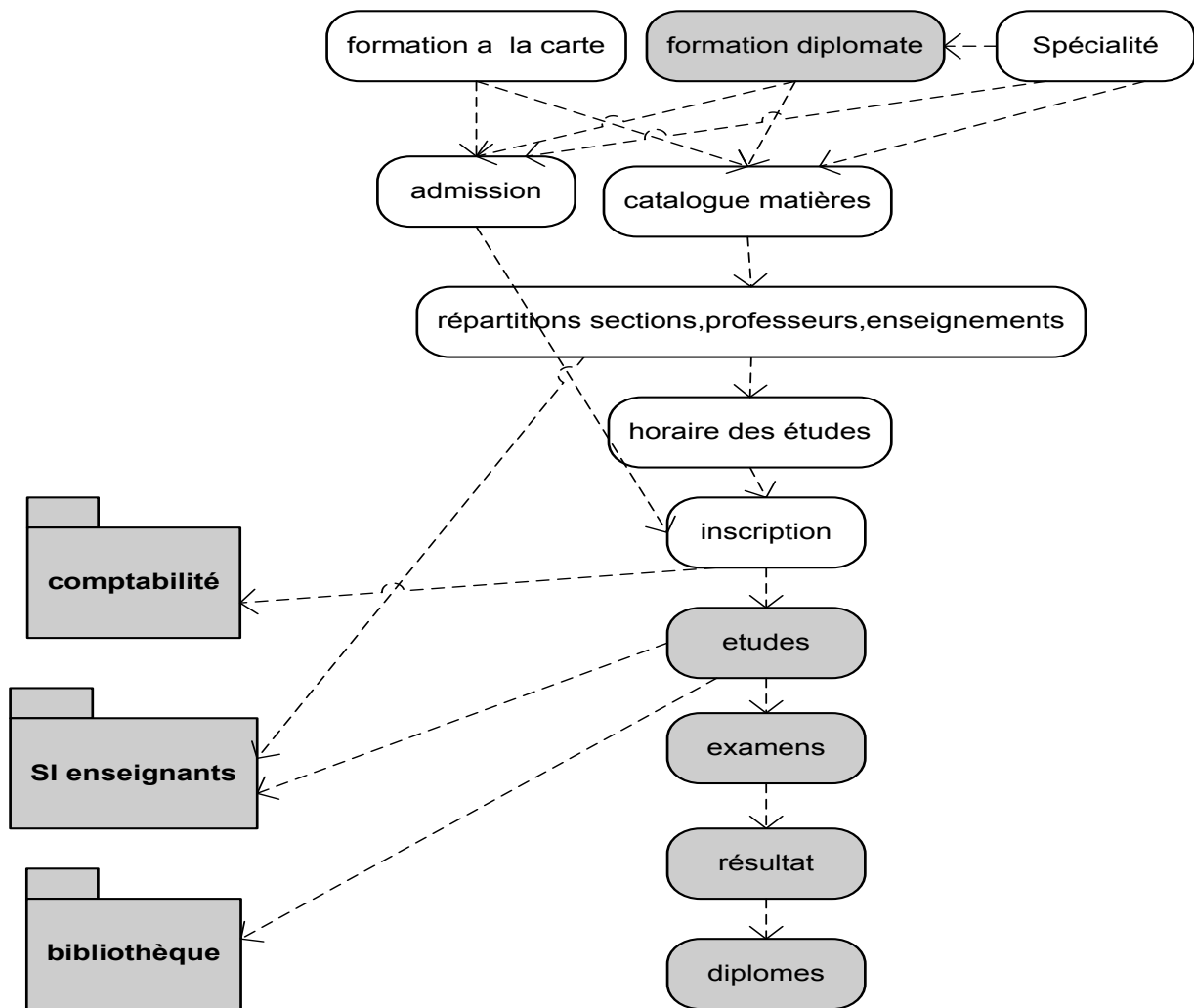


Figure 28: Première version du SI ISAE-Cnam Liban

VII.2.3 Deuxième version

La deuxième version est délivrée durant l'année universitaire 2009-2010, elle couvrait 90% du système d'information des étudiants. Cette version est schématisée par le graphe si dessous, les parties en couleur grise ne sont pas encore terminées à la deuxième version.[14]

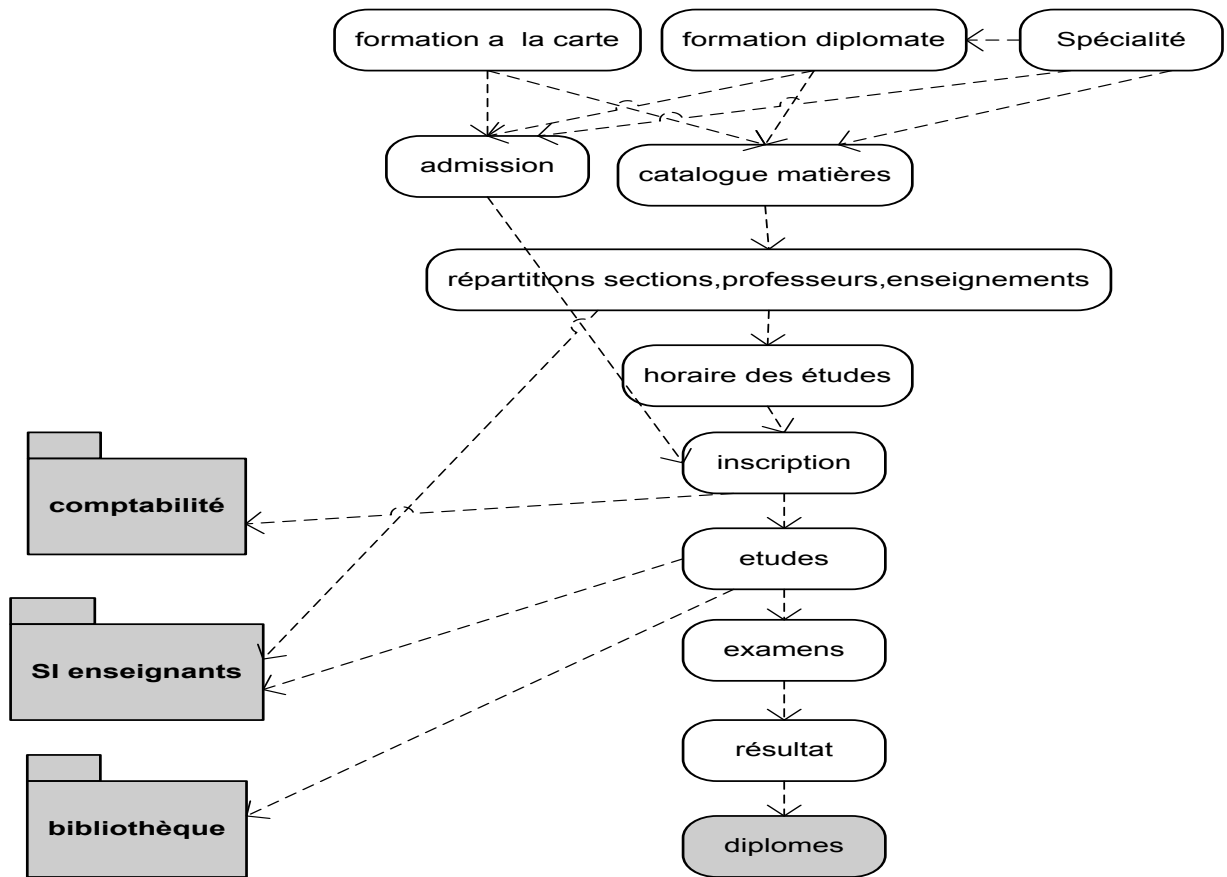


Figure 29: Deuxième version du SI ISAE-Cnam Liban

VII.3 Description détaillée

VII.3.1 Module SI Étudiants

Les principaux services du module SI-Étudiants sont :

- Inscription
- Préinscription
- Catalogue des diplômes
- Equivalence des matières
- Admission
- Accord chef
- Horaire
- Profile de l'étudiant
- Résultat

- Statistique sur les résultats
- Email
- Stage pour les étudiants
- Sécurité sociale

VII.3.1.1 Description du module « préinscription/inscription »

Le module « préinscription/inscription » est la partie essentielle du SI de l'ISAE-Cnam Liban, c'est l'application centrale du SI. L'inscription des étudiants aux matières (ou aux crédits) est précédé par une étape de préinscription, où les étudiants réservent leurs matières sur internet sans besoin de se présenter personnellement à l'institut. Durant l'étape de préinscription, la direction de l'institut fait une analyse sur les nombres des étudiants préinscrits aux matières ouvertes dans un semestre donné, en vue de décider quelles sont les matières qui doivent être enseignées pendant le semestre courant et fermer les autres dont le nombres des étudiants préinscrits est faible. Voilà l'interface de l'inscription où un étudiant est inscrit dans l'ensemble des matières en couleurs jaunes [Figure 30], de plus vous pouvez visualiser l'automate de préinscription et d'inscription, qui se débute par la préinscription et se termine par le paiement de la facture d'inscription [Figure 31] :

[Gestion dossier](#)
[Informations de base](#)
[Informations ISAE](#)
[Informations de contact](#)
[Sécurité Sociale](#)
[Matières résumés](#)
[Matières dispensées](#)
[Matières suivies](#)
[Notes 2009](#)
[Etats demandes](#)
[S'inscrire](#)
[Stats](#)

Spécialité :

Diplôme :

Titre RNCP - Niveau III Analyste programmeur
 Titre RNCP - Niveau II Concepteur - architecte informatique
 Diplôme d'ingénieur CNAM (HTI) Spécialité Informatique - option Réseaux, systèmes et multimédia (RSM)

Votre panier contient :

Sélectionner	Nom matière	Code	Crédit	Prioritaire	NoOuverture	Période	Spécialité
☐	Français Niveau N0 (Découverte)	LB010	0	Oui	1	--	Electrotechnique
☐	Commande des systèmes à événements discrets	AUT103	6	Oui	1	--	Electrotechnique
☐	Bases de transmissions numériques (1)	ELE112	6	Oui	1	--	Electronique
☐	Technologie des Hauts Débits	ELE207	6	Oui	1	--	Electronique
☐	Français Niveau N1 (2) (Intermédiaire)	LB020	0	Non	1	--	Electrotechnique

N.B. : Cliquez sur le bouton "Etape2/2 - OK" pour que l'inscription soit faite.

Figure 30: Préinscription

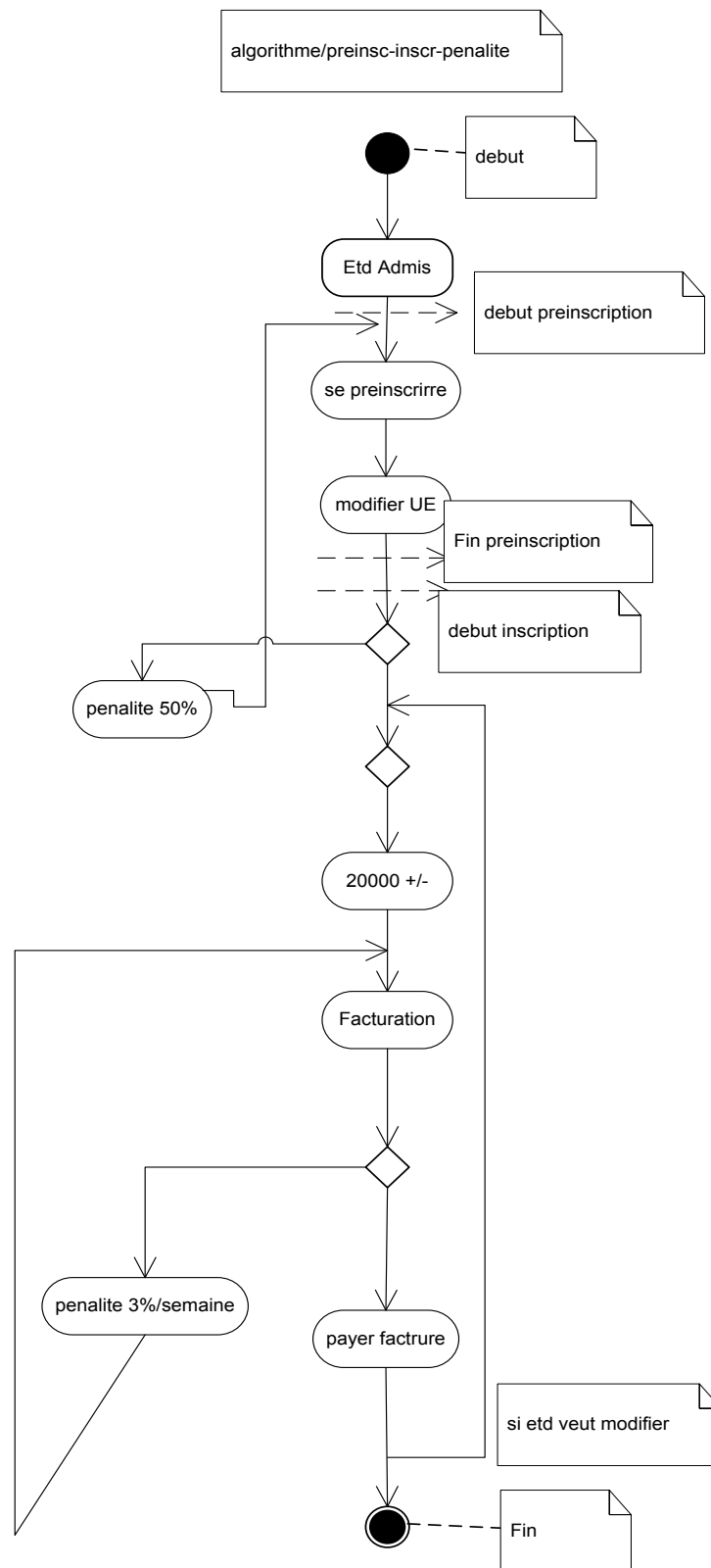


Figure 31: Automate d'inscription

VII.3.1.2 Description du module « Horaire des Etudes »

L'horaire des études est la répartition des enseignements en fonction du temps (jour/période). La production de l'horaire des études s'effectuent par le chef de département, chaque horaire correspond à :

- Année universitaire.
- Centre d'enseignement.
- Spécialité.
- Gradeaff (nombre du -1 à 14 qui correspond à l'année de parcours du diplôme et au semestre).

La partie horaires des études est composées de deux sous parties sont les suivantes :

a- L'application « HoraireIsae »

Cette application possède deux versions, la première est adressée au Chef de département (lecture/écriture) et la deuxième adressée au publique (lecture), grâce à la première version l'utilisateur privilégié peut :

- Insérer ou visualiser l'horaire.
- Afficher les conflits des séances d'enseignement d'un professeur (pour les mêmes jour/période).
- Imprimer des rapports de l'horaire.

L'approche de la programmation est orientée composant, c'est-à-dire que la page principale [Figure 32] contient trois pages JSP ; chaque JSP a un rôle spécifique comme l'indique l'interface principale suivante :

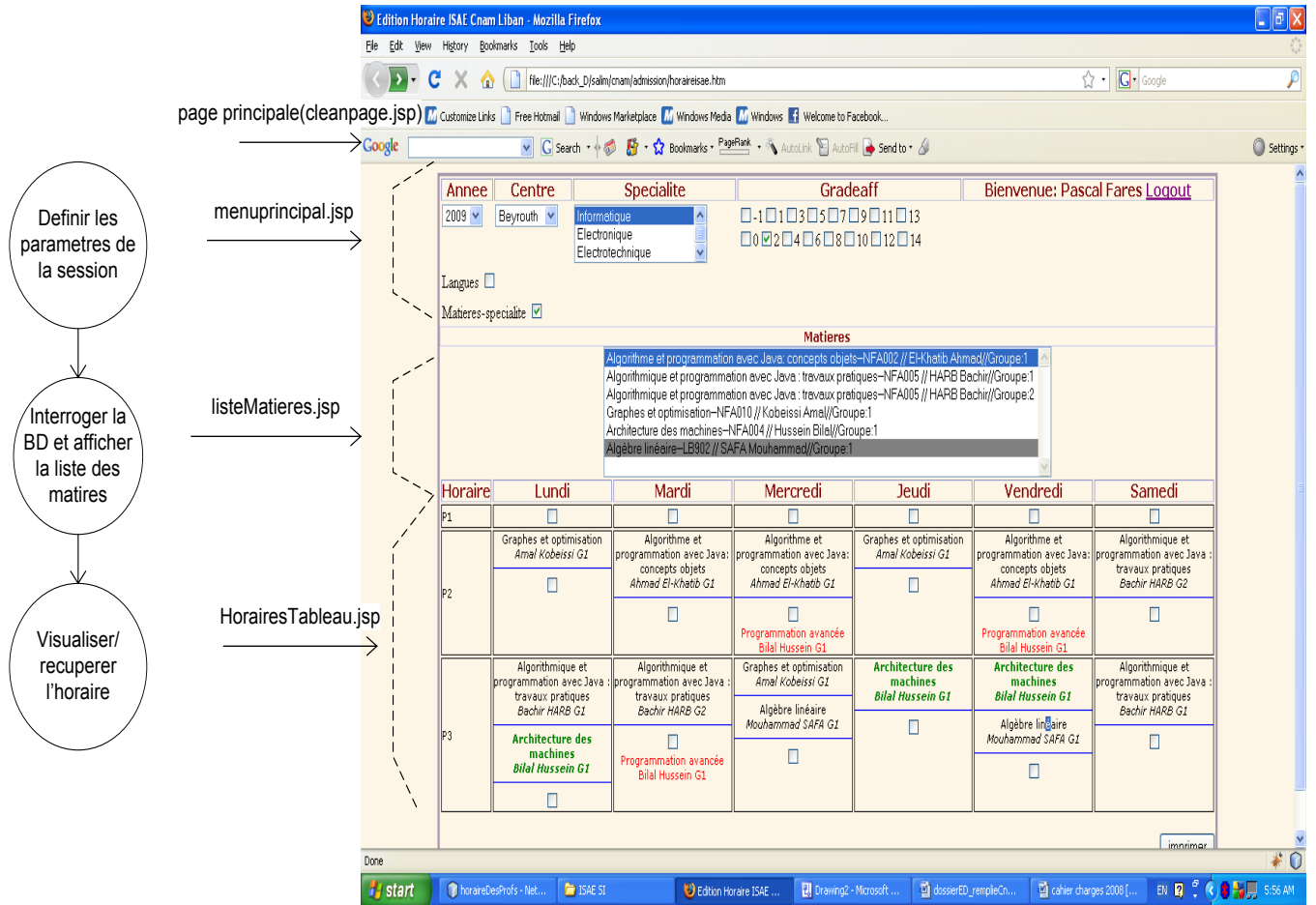


Figure 32: Gestion Horaire

b- Les applications : Horaire d'étudiant, Horaire du professeur, Horaire du chef :

Les applications mentionnées ci-dessus, fournissent l'horaire du demandeur (étudiant, professeur, chef). Ce sont trois projets qui utilisent le même composant, qui s'appelle « interface horaire ». Cette méthode de programmation orientée composant nous donne l'avantage qu'un projet web peut être un composant réutilisable par plusieurs autres projets comme l'indique la figure 11 , section [III.4.6].

Aussi, il y a de diverses applications comme le stage, les résultats, les services d'emails, l'interface chef, je ne veux pas ici expliquer toutes les applications mais seulement les essentielles. Le schéma suivant représente l'interface du «Stage » avec les variables qui sont bien lisibles, en vue pour inscrire un étudiant dans un stage :

STAGE:

Salim Antoine RAHAL

Tuteur actuel: Bourhane Dabboucy

Date début : 2009-11-30 * yyyy-mm-dd

Date fin : 2005-09-09

Durée en mois: 2 *

Type : Privé

Nom Entreprise : wewe

Etabliss. Public : —

Présence Justificatif: ☒

Payé : ☒

Dispensé : ☐

Terminé : ☒

Spécialité: Informatique *

Figure 33: Stage

Prérequis et Conseil Substitution Accord Chef Informations Etudiant Admission Dossier Edition Horaire Pré-Inscription/matière Inscription/matière Catalogue matière Modifier Mot de Passe	Demandes Demandes refusées Liste des accords				
	Liste des accords:				
No Dossier	Etat		Nom	Code CNAM	Nom matière
2506 f	Etudiant inscrit		CHAMOUN Nassib Mounir	ENG200	L'Ingénieur au XXIème Siècle
2506 f	Etudiant inscrit		CHAMOUN Nassib Mounir	ENG110	Management de Projet pour l'Ingénieur
4791 f	Accordée	Refus	ABI RACHED Walid Maroun	ENG110	Management de Projet pour l'Ingénieur
7059 f	Accordée	Refus	IBRAHIM Helena Gerges	ENG200	L'Ingénieur au XXIème Siècle
351 bak	Accordée	Refus	AZZAM Sara Youssef	RSX102	Technologies pour les applications client-serveur
468 S	Accordée	Refus	AMACHA Maysaa Ali	NSY104	Architecture des systèmes informatiques
6604 f	Accordée	Refus	AHMAD Ali Sami	NFP111	Systèmes et applications répartis
614 p	Accordée	Refus	KADY Elie Farah	RSX103	Réseaux : Compléments et applications
36 BAAL	Accordée	Refus	HAYDAR A'laa Hani	ENG200	L'Ingénieur au XXIème Siècle
4124 f	Etudiant inscrit		NAIM Manal Ali	UA5M14	Préparation, rédaction et soutenance du mémoire
5133 F	Etudiant inscrit		ZOAYTER Minerba Ahmad	UA5M14	Préparation, rédaction et soutenance du mémoire
3932 f	Etudiant inscrit		EL KHOURY Pierre Elias	UA5M14	Préparation, rédaction et soutenance du mémoire
801 t	Accordée	Refus	AL-BOURHAN Bassam Aref	NFP111	Systèmes et applications répartis
7371 f	Etudiant inscrit		TARTARIAN Paul Elie	RSX103	Réseaux : Compléments et applications
7371 f	Etudiant inscrit		TARTARIAN Paul Elie	NFP111	Systèmes et applications répartis
88 BAAL	Accordée	Refus	CHAMAS Samar Hassan	ENG110	Management de Projet pour l'Ingénieur
6639 f	Accordée	Refus	MAHMOUDI Linda Ahmad	NFP210	Construction rigoureuse des logiciels(2)
1512 A'	Accordée	Refus	TABANJI Tanios Michel	ENG200	L'Ingénieur au XXIème Siècle
7880 f	Etudiant inscrit		CHREIF ali Yehya	RSX103	Réseaux : Compléments et applications
6255 f	Etudiant inscrit		RIDA Linda Rida	NFP210	Construction rigoureuse des logiciels(2)
6255 f	Etudiant inscrit		RIDA Linda Rida	ENG200	L'Ingénieur au XXIème Siècle
315 bak	Etudiant inscrit		ZEINEDDINE Naji Bachir	NSY104	Architecture des systèmes informatiques
3511 F	Accordée	Refus	HADDAD Elie Karim	NFP121	Programmation avancée
3511 F	Accordée	Refus	HADDAD Elie Karim	RSX102	Technologies pour les applications client-serveur
3511 F	Accordée	Refus	HADDAD Elie Karim	NSY104	Architecture des systèmes informatiques

Figure 34: Interface chef (liste des accords)

VII.3.2 Module SI Enseignants

les applications du SI-Enseignants sont :

- Présence des professeurs
- Salaire et contrat des professeurs
- Coordonnées des professeurs
- Horaire des professeurs

Notons, que la partie salaire et contrat des professeurs n'est pas encore programmée en JAVA , elle est actuellement sur ACCESS. Nous sommes en train de migrer les applications prioritaires avant les autres.

VII.3.3 Contribution personnelle

Le projet du système d'information de l'ISAE-Cnam Liban, est constitué de six modules cités dans la partie «VII.2 .1 Les Modules». L'équipe du développement est constitué de cinq personnes : un chef du projet, un chef de développement et deux développeurs, dont moi-même. Actuellement nous sommes en train de réaliser la deuxième version du SI.

J'ai contribué comme développeur et analyste à la réalisation de la SI Étudiants et Enseignants, cette contribution comporte :

- Choix de l'architecture adaptée : Modèle-Vue-Contrôleur (MVC)
- Participation à l'analyse et la conception en utilisant la méthode de modélisation UML
- Programmation des sous modules
- Tests et déploiements
- Tâches de configurations des systèmes (bases de données : réplication etc.)

VIII Développement des rapports

VIII.1 Recherche d'un outil des rapports

Au début du développement du nouveau SI, les rapports sont développés en HTML. Le développement des rapports en HTML était un travail difficile, et nécessite beaucoup du temps, il était nécessaire de chercher un outil qui facilite la manipulation des rapports. Cet outil envisagé, devra être compatible avec l'architecture JAVA/JSP/SERVLET, car notre objectif est de produire des rapports sur Web, autrement dit il faudra connecter nos rapports aux applications Web, et pouvoir transférer des paramètres entre eux. La communauté SUN a proposé pour l'outil de développement « NetBeans6.5 », un outil des rapports qui s'appelle « JasperReport - Plugin », mais cet outil pose beaucoup de problèmes de conflits et de cohabitation avec l'outil de développement, donc ce choix était éliminé. Après les recherches sur le Web, j'ai trouvé un outil populaire qui s'appelle « IReport », qui est une version améliorée de l'ancien module du « NetBeans », cette version est un logiciel complet et indépendant du « NetBeans », et possède un support par la communauté « SourceForge.org », le véritable équipe d'Open Source.

VIII.2 Architecture du « Jasper Report »

« Jasper Report » [Figure 35] est constituée des parties suivantes :

- Moteur « Jasper Report »
- Compilateur « Jasper Report »
- Variétés des outputs:
 - PDF, HTML, XLS, CSV, XML, RTF, TXT
- Variétés des connexions (« data source ») :
 - JDBC, EJB, Hibernate, Java Bean...

VIII.3 « IReport » (Designer)

Le « IReport » peut être encadrer par les points suivants :

- Outil de design visuel.
- Dernière version 3.7.X.
- capable de gérer le design des rapports et les compiler.
- Contient un éditeur des expressions(condition?<exp>:<exp>....).

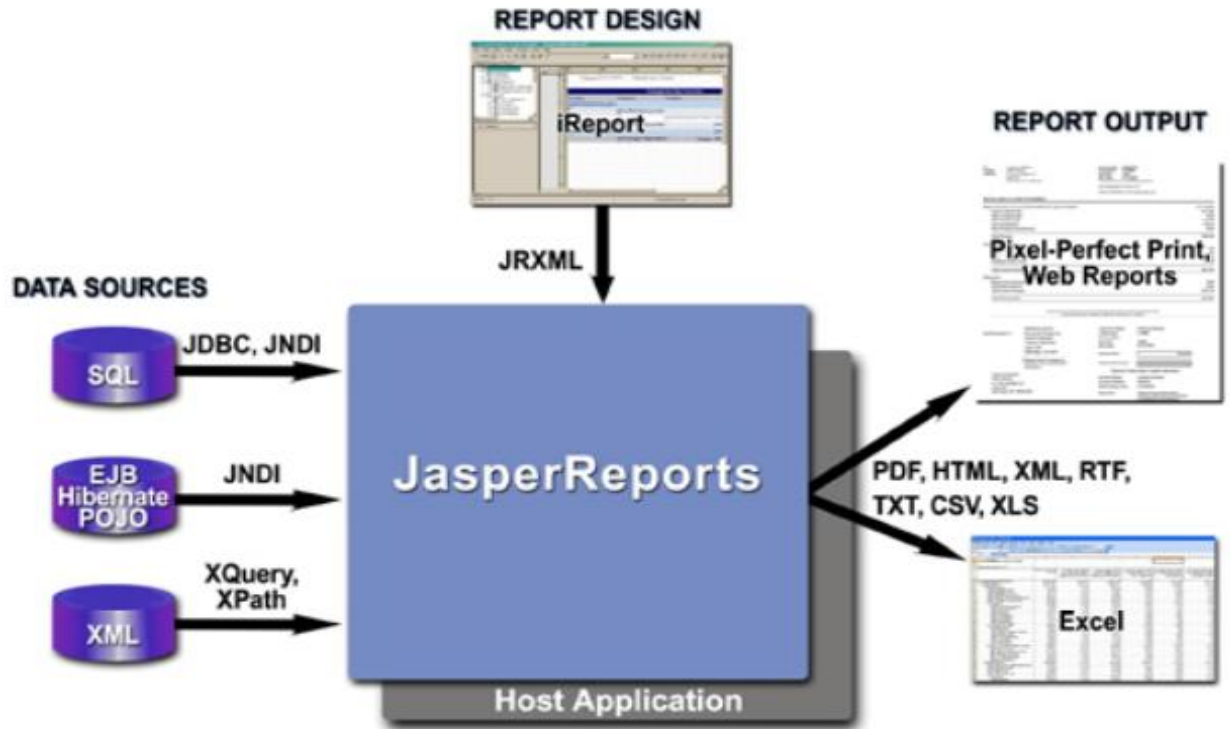


Figure 35 : Architecture « Jasper Report »

VIII.4 Les fichiers

Deux types de fichiers sont créés par « IReport » :

- .Jrxml – Définition d'un rapport (Xml format).
- .Jasper – Rapport compilé

Le fichier « .Jrxml » peut être édité directement par le développeur ou automatiquement par le « IReport », ce fichier représente un rapport en format Xml, quand le fichier est compilé le « .Jasper » est créé. Notons que « IReport » peut ouvrir les deux fichiers pour les éditer.

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport>
  <parameter name="annee" class="java.lang.String"/>
  <queryString>
    <![CDATA[select t1.numprof,t1.prenomprof,t1.nomprof from db_isae.Professeur as
t1,db_isae.HoraireHebdo as t2,
db_isae.Enseignement as t3, db_isae.Section as t4
where t3.numprof=t1.numprof and t3.noenseig=t2.noenseig and t3.NoSection=t4.NoSection
and
t4.annee=$P{annee} and t4.IdCentre=$P{idcentre} and NoOuverture=$P{noouverture} and
t2.jour=$P{jour}]]>
  </queryString>
```

Figure 36: Fichier JRXML

VIII.5 Intégration « Jasper » avec Web/MVC

Le modèle MVC nous a permis d'intégrer des rapports compilés .Jasper à nos applications Web. L'élément base de cette intégration est le « Servlet », puisqu'il possède des sorties divers, comme HTML, Pdf, Excel, Rtf etc.

Après des études approfondies en « Jasper Report », j'ai pu cohabiter les rapports Jasper avec le Web, mais beaucoup de difficultés sont rencontrées au début, citons à titre d'exemple:

- Comment gérer les connexions à la base de données, concernant les rapports et les application dont ils sont rattachés ?
- Quelles librairies faut-il utiliser pour les applications, en vue de générer proprement les rapports ?
- Quelle sortie (Html, Pdf, etc.) faut-il utiliser dans le « Servlet » ?
- Est-ce qu'il faut déployer le rapport avec l'application Web ou indépendamment ?

Grâce à la recherche sur l'internet, et après de tests rigoureux, j'ai pu opérer cette architecture, et répondre sur les questions précédentes :

- La connexion du rapport doit être passée en paramètres via l'application principale.[Figure 37]
- J'ai utilisé le minimum des librairies qui peuvent opérer le logiciel, comme le « JasperReport.Jar » et « IText.Jar ».
- Les rapports de type Pdf sont les meilleurs, dont le choix est tombé sur Pdf.[Figure 38]
- Il faut déployer les deux : le rapport et l'application, en déterminant le chemin du rapport dans l'application, pour être accessible à l'application.

```

//obtenir la connection
Connection connection = (Connection)
pack1.StaticClass.establishConnection(saisieBarcode.StaticClass.url,
saisieBarcode.StaticClass.user, saisieBarcode.StaticClass.password);
// fill the report
JasperPrint jasperprint = JasperFillManager.fillReport(
    saisieBarcode.StaticClass.reportPath + rptfilename, params,
    connection);
byte[] pdfasbytes = JasperExportManager.exportReportToPdf(jasperprint);

ServletOutputStream outstream = response.getOutputStream();
response.setContentType("application/pdf");
response.setContentLength(pdfasbytes.length);
response.setHeader("Content-disposition",
    "inline; filename=\"Report.pdf\"");
outstream.write(pdfasbytes);

```

Figure 37: « Servlet » (générateur Jasper->Pdf)

Numero	Prenom	Nom	Arrivee	Depart	Notes
205	Bachir	HARB			
22	Noureddine	Assaad			
156	Khalil	Zeineddine			
609	Mohamad	Alwan			

Figure 38: Rapport.pdf

IX Bilan du projet

IX.1 État d'avancement

Le début effectif du développement du projet de l'ISAE-Cnam Liban était en septembre 2008, après 30 mois de travail, c-à-dire jusqu'à mars 2011, 50% du projet était réalisé. La partie achevée durant les deux années dernière couvre 90% du SI de l'étudiant et 90 % du SI de l'enseignant, les parties restantes sont les sous modules de répartition des classes pour les examens et les contrats des enseignants. Aussi les parties suivantes ne sont pas encore touchées :

- Ressources humaines.
- Comptabilité.
- Stock.
- Bibliothèque.

D'autre part, n'oublions pas que durant le développement des nouveaux logiciels, nous effectuons des tâches de maintenance et d'amélioration des logiciels déjà programmés, par exemple au cours du développement du module de l'examen, des améliorations sont effectuées sur l'affectation des matières. Par conséquent, je prévois achever la clôture du projet après 2 ans (mars 2013), en complétant les autres modules non réalisés et améliorer les autres déjà développés. Notons ici, que cette durée comporte aussi des travaux administratifs, comme la surveillance des examens etc.[15]

IX.2 Problèmes rencontrés

IX.2.1 Problèmes fonctionnels

Tout logiciel pose de problèmes, surtout au début d'utilisation. Beaucoup de problèmes sont rencontrés, par exemple dans le cas de l'étudiant libre qui vise choisir seulement des matières sans obtenir un diplôme, notre logiciel jusqu'à maintenant applique le système des prérequis pour tous les étudiants sans distinguer entre un étudiant libre et autre visant un diplôme. Autre cas erroné est détecté après le lancement de la préinscription, ce qui concerne les contrats des enseignants, ces contrats doivent être réalisés avant l'inscription des étudiants dans les matières c'est-à-dire au début de cette période, or ce n'est pas le cas, car pas tous les professeurs sont déterminés dans le délai convenable, or comment on peut manipuler l'horaire d'un professeur non encore déterminé ? on a abouti à une solution temporaire, c'est choisir un professeur « X » imaginaire dans le but de réaliser les affectations des matières et l'horaire du professeur. Mais ce n'est pas possible de populer un contrat pour un professeur « X », il faut attendre le professeur réel !

Citons aussi un autre cas rencontré dans la période de l'inscription, si le délai entre la date de paiement de l'étudiant chez Liban Post et la réception des listes des étudiants payées par nous

est assez long, il y a un risque de tomber dans la faille suivante : si un étudiant déjà inscrit demande son attestation de notes et le système n'a pas encore confirmé sa inscription, sa demande pourra être refusée, car il est considéré non inscrit. En effet, nous avons demandé du Liban Post de passer les listes chaque deux jours, pour minimiser ce cas erroné.

IX.2.2 Problèmes techniques

Ce paragraphes traite certaines erreurs (« bugs ») techniques durant les phases de configuration, du développement, du test et d'exécution, enfin du déploiement, citons à titre d'exemple :

- Phase de configuration: configurer des rapports Jasper pour supporter l'arabe était une opération assez difficile au début, mais après, nous avons trouvé la solution ! maintenant, les rapports en arabes fonctionnent normalement sur le serveur Linux et sur ma machine, mais les autres postes occupés par Windows peuvent seulement télécharger les rapports en arabe, mais ne peuvent pas les programmer, c'est un problème lié au système Windows. Aussi il y a une autre faille liée au Windows, ce sont les attaques des virus. Personnellement, je préfère utiliser Linux pour le développement et Windows pour autres tâches.
- Phase de développement : Pour les applications qui intègrent AJAX, certaines de ces applications ne fonctionnent pas sur le navigateur « Internet Explorer », elles fonctionnent seulement sur « Mozilla FireFox » et vice-versa. En effet, ce problème est liée aux méthodes JavaScript utilisées par AJAX, chaque navigateur possède ses propres méthodes d'accès à son schéma DOM (Document Object Model).
- Phase de test et d'exécution: La plupart des erreurs ou des bugs, sont découvertes au moment du test et d'exécution. Les erreurs les plus dangereuses, sont les intelligentes et les transparentes au développeur. Par exemple durant une transaction donnée, il y avait des insertions erronées à la base de données dues à une faute de programmation, ce type d'erreurs pourra être détecté après l'analyse des informations insérées à la base de données durant une certaine période.
- Phase de déploiement: Des erreurs sont détectées durant et après le déploiement au serveur d'application, par exemple une connexion non fonctionnelle à la base de données, une application Web ne peut pas être lancée par le serveur etc.

IX.2.3 Temps de développement

Au début du projet, les configurations nécessitaient beaucoup du temps, mais 80% des configurations sont réalisées une seule fois, par exemple les configurations du MySQL et du serveur Apache Tomcat, par contre le reste du 20% peut être considéré comme un temps des tâches répétitives (par exemple changer la connexion à la base de données) et de poursuivre des versions et de mise à jour. D'autre part, le temps de développement pour réaliser la version

actuelle du logiciel était à peu près 35% du temps globale, bien sûr ce temps dépendait des algorithmes, car certains algorithmes nécessitaient beaucoup du temps, comme l'application des résultats, qui est constituée des algorithmes complexes, et utilisait une librairie puissante qui converties les formules du syntaxe au sémantique, développée par le chef du projet. En effet, il y a une relation fondamentale entre le temps d'analyse et le temps du développement, si le temps d'analyse et du conception augmentent, alors le temps du développement diminuera, à condition que l'analyse et la conception sont bien étudiés.

IX.2.4 Amélioration possible

Nous pourrons améliorer notre SI suivant deux axes, l'un fonctionnel et l'autre technique. Fonctionnellement, il faudra consacrer plus du temps pour l'analyse et la conception pour certain module (SI Étudiant par exemple), par conséquent nous pourrons minimiser le temps du développement, et les erreurs fonctionnelles citées précédemment.

Techniquement, les tests qui touchent tous les boucles et les conditions, en analysant les traces du programme, ce sont des tâches qui peuvent être suffisantes pour détecter les erreurs intelligentes, mais n'oublions pas de visualiser les données affectées et les changements réalisés au niveau de la base de données. En plus, la présentation ou le «Design» de nos applications nécessite d'amélioration, surtout des nouveaux style CSS sont meilleurs pour rendre nos pages d'accueilles plus attrayantes et professionnelles.

Conclusion

Le Système d'Information de l'ISAE-Cham Liban est en cours de développement; la première étape du projet est achevée en développant le SI des étudiants et le SI des enseignants, mais ces deux grands modules nécessitent encore des améliorations. Entre temps, nous sommes en train d'ajouter certaines nouvelles fonctionnalités à ces modules, comme l'ajout du module « SuiviÉtudiant » pour le SI des étudiants, qui gère les différents rapports que l'étudiant demande. La seconde étape prévue consiste à réaliser la gestion de la comptabilité, la gestion de stock et la bibliothèque, dont je prévois délivrer dans deux ans. Rappelons que la première étape était finie en deux ans.

Ce nouveau système possède des nouvelles caractéristiques qui sont :

- Modulaire
- Architecturale
- À jour
- En ligne

La modularité est exprimée par la POC et l'architecture est implémentée dans le modèle « Model-View-Controller ». D'autre part, notre système est toujours à jour car nos serveurs comme la JVM, le serveur « Apache-Tomcat », la base de données MySQL, l'outil de développement « NetBeans » et l'outil des rapports « IReport » qui subissent des mises à jour périodiques.

Par ailleurs, dans mon mémoire j'ai montré que la réalisation d'un logiciel ne nécessitait pas seulement de la technique, mais aussi de la gestion et de la communication, car pour comprendre les besoins exacts d'un client, nous devons être au courant des travaux administratifs, donc il faut « vivre le cas » et l'ambiance de la société, pour être proche de ses besoins et ses difficultés, en vue de développer la meilleure solution.

En effet, ma participation au développement du SI, m'a donné le droit de poser les deux questions suivantes :

- Comment pourrions-nous préserver « la valeur » de notre logiciel par rapport à nos clients ou à notre administration, en augmentant sa durée de vie?
- Après dix ans par exemple, comment pourrions-nous « comprendre » les programmes et faire de la maintenance du code?

Désormais, je pourrais garder « la valeur » du logiciel réalisé tout en satisfaisant les demandes infinies de la direction, surtout que notre SI doit représenter notre institut sur le plan informatique et ce SI doit être flexible et adaptable aux nouveaux besoins. Concernant la maintenance, diverses opérations peuvent la faciliter comme l'ajout des commentaires aux codes, l'arrangement du code d'une manière lisible et la composition en modules avec des paramètres d'échanges etc..

Enfin, la sécurité est un élément essentiel pour protéger un SI, par ailleurs l'infrastructure du réseau à l'institut a besoin de l'amélioration pour faciliter certaines tâches, et surtout la

sécurité. Le réseau virtuel privé offre des services qui facilitent l'accès à nos postes de travail au domicile et de relier les centres d'enseignements au VPN d'une manière crypté . Un logiciel gratuit qui s'appelle OPENVPN existe au marché, il est populaire, et peut être intégré aux machines Windows et Linux. Concernant la sécurité des applications Web, il y a la réécriture des URL, en vue de masquer le vrai URL et les paramètres envoyés; cette fonction est implémentée dans le serveur Apache. Mais le développement du SI nous a empêché jusqu'à maintenant de lancer ces mini projets.

Bibliographie

Références électroniques

[1] Open Source. In : Wikipédia.org, [en ligne].

Disponible sur : < http://fr.wikipedia.org/wiki/Open_source >.(consulté le 26-9-2010)

[2] Le marché de l'open source vise les 6 milliards de dollars pour 2011. In : L'atelier.fr, [en ligne].

Disponible sur :

< <http://www.atelier.fr/informatique/marche,open,source,vise,6,milliards,dollars,2011-34557-26.html> > . (consulté le 20-9-2010)

[3] The Linux Kernel Archives. In kernel.org, [en ligne].

Disponible sur : <<http://www.kernel.org>>.(consulté le 21-9-2010)

[4] Shields UP Services. Gibson Research Corporation. In grc.com, [en ligne].

Disponible sur : <<https://www.grc.com/x/ne.dll?bh0bkyd2>>.(consulté le 21-9-2010)

[5] Java. In : Oracle. Java.com,[en ligne].

Disponible sur :< <http://www.java.com>>

[6] Le comparatif détaillé 01net./Novaforge. In : 01netPro ,[en ligne].[publié le 27/02/2009].

Disponible sur :

< <http://pro.01net.com/editorial/404092/le-comparatif-detaille-01net-novaforge>>.(consulté le 26-9-2010)

[7] Programmation orienté objet. In : Wikipédia.org, [en ligne].

Disponible sur :< <http://fr.wikipedia.org/wiki/POO>>.(consulté le 1-9-2010)

[8] Programmation orientée composant. In : Wikipédia.org, [en ligne].

Disponible sur :

< http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_composant>.(consulté le 1-9-2010)

[9] MySQL : les principales différences entre MyISAM et InnoDB. In : Tux-planet. Tux-planet.fr, [en ligne].

Disponible sur :< <http://www.tux-planet.fr/mysql-les-principales-differences-entre-myisam-et-innodb/>>.(consulté le 1-9-2010)

[10] AJAX. In : Linux-mag.fr, [en ligne].

Disponible sur :< <http://www.linux-mag.fr/?ref=ajaxarticle>>.(consulté le 2-9-2010)

[16] Microsoft Access. In : Wikipédia.org, [en ligne].

Disponible sur :< http://en.wikipedia.org/wiki/Microsoft_Access>.(consulté le 11-10-2010)

Références imprimées

- [11] TARTARIAN N., 2008. Cahier de charges de SI de l'ISAE-Cnam Liban. 87 p.
- [12] Chromatic, 2005. Extreme programming. O'Reilly. 100 p.
- [13] PITMAN N., 2006. UML 2 en concentré. O'Reilly. 200p.
- [14] ROCUQUES P. et VALLÉE F., 2004. UML 2 en action. 3e édition. Eyrolles, France, 386 p.
- [15] GORON Peter. Développement d'un MRP à capacité finie pour l'ERP libre OfbzNeogia. Rapport de projet de fin d'étude. Tours : Ecole polytechnique de l'Université de Tours, 2005. 75 p.

Table des annexes

Annexe 1 Servlet – Application « sécurité sociale ».....	109
Annexe 2 Classe « UtilFunctions »	112
Annexe 3 Filter – Application « HoraireIsacTC »	114

Annexe 1 Servlet – Application « sécurité sociale »

```
/**
 *
 * @author salim
 */
public class SSServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException, SQLException, NamingException {
        int noetud = 0;//26051
        int annee = 0;
        int semestredebut = 0;
        int numattestation = 0;
        int typesecurite = 0;
        int nossanc = 0;
        String msg = "";
        String address = null;
        String y = "";
        pack1.Fiche f = null;

        String value = request.getParameter("submit");
        String noetudstr = request.getParameter("noetud");
        String anneestr = request.getParameter("annee");
        String noss = request.getParameter("noss");

        String numattestationstr = request.getParameter("numattestation");
        String typesecuritestr = request.getParameter("typesecurite");
        String semestredebutstr = request.getParameter("semestredebut");
        String nossancstr = request.getParameter("nossanc");// l'ancien noss
        if (StaticClass.debug) {
            System.out.println("++SSServlet:7 valeur recues:submit :"+ value + "/etud" + noetudstr + "/annee" +
anneestr + "\n");
            System.out.println("/noss" + noss + "/numattestation" + numattestationstr + "/typesecurite" +
typesecuritestr + "/semestre debut" + semestredebutstr);
        }
        if (value != null) {

            if (noetudstr != null && anneestr != null && semestredebutstr != null && typesecuritestr != null) {

                noetud = Integer.parseInt(noetudstr);
                annee = Integer.parseInt(anneestr);
```

```

semestredebut = Integer.parseInt(semestredebutstr);
type securite = Integer.parseInt(type securitestr);

//definir les variables annee,noetud pour preserver la session
HttpSession session = request.getSession();
Bean bean = (Bean) session.getAttribute("bean");
if (bean != null) {
    bean.setAnnee(annee);
    bean.setNoetud(noetud);
} else {
    msg = "erreur--bean est null";
}
session.setAttribute("bean", bean);

if (value.equalsIgnoreCase("submit")) {
    if (semestredebut != 0 && type securite != 0) {
        //1-1 cas ou on a nouveau ss
        try {
            if (StaticClass.debug) {
                System.out.println("SSServlet....attend de creation d'un compte");
            }
            ssPack.StaticMethods.CreerCompteSecTemp(noetud, annee, semestredebut, type securite);
        } catch (SQLException ex) {
            Logger.getLogger(SSServlet.class.getName()).log(Level.SEVERE, null, ex);
            if (StaticClass.debug) {
                System.out.println(ex.getLocalizedMessage());
            }
        }
    }

    if (noss != null) {

        /*//travail sur le 10.130.0.12
        noss = new String(request.getParameter("noss").getBytes("ISO-8859-1"), "UTF-8");
        y="avant contient Ø";
        //si le nom n'est pas decode normalement tester pour:é è à
        if (noss.contains("Ø")) {
            noss = request.getParameter("noss");
            y="noss.contains(Ø)";
        }*/

        if (!noss.isEmpty()) {
            if (noss.length() > 12) {
                msg = "erreur-- No. SS : Max 12 character";
            } //valider compte
            else {
                ssPack.StaticMethods.ValiderCompteSec(noetud, annee, noss);
            }
        }
    }
}

//1-2 cas ou on a ancien ss-->nouveau
if (nossancstr != null) {
    //ici je veut valider l'ancien ss sur la nouvelle annee

```

```

        if (!nossancstr.isEmpty()) {
            //avoir secanc puis inserer le noss ancien dans le nouveau
            Securite secanc = StaticMethods.AvoirSec(noetud, annee - 1);

            if (secanc != null) {
                //msg += secanc.getAnnee();
                StaticMethods.ValiderCompteSec(noetud, annee, nossancstr);
            }
        }
    }
    if (numattestationstr != null) {
        if (!numattestationstr.isEmpty()) {
            try {
                numattestation = Integer.parseInt(numattestationstr);
            } catch (NumberFormatException numberFormatException) {
                msg = "erreur-- Numero d'attestation doit etre un Nombre";
                if (StaticClass.debug) {
                    System.out.println(numberFormatException.getLocalizedMessage());
                }
            }
            if (numattestation != 0) {
                ssPack.StaticMethods.InsererNumAttestation(noetud, annee, numattestation);
            }
        }
    }

    } //fin if numattestation
} //fin semestre type sec not null
else {
    msg = "erreur-- semestre debut(ou type securite) : Valeur Non Valide";
}

    // msg+="/noss"+noss+y;
    address = "ss.jsp?msg=" + msg;
} //fin de if submit
//le bouton delete est deactive
else if (value.equalsIgnoreCase("delete")) {
    //ici on fais forward to cons
    address = "confss.jsp?noetud=" + noetud + "&annee=" + annee;
}

} //end noetud annee..not null

} //end if value not null

RequestDispatcher dispatcher = request.getRequestDispatcher(address);
dispatcher.forward(request, response);
}

```


Annexe 2 Classe « UtilFunctions »

```
/**
 *
 * @author salim
 */
public class UtilFunctions {

    //hash<string,string>==>List<string,string>
    public static List HashtoList(Hashtable ht) {

        List list = null;
        if (!ht.isEmpty()) {
            list = new LinkedList();
            Enumeration enum = ht.keys();
            while (enum.hasMoreElements()) {
                String key = (String) enum.nextElement();
                String value = (String) ht.get(key);
                KeyValueObj o = new KeyValueObj(key, value);
                list.add(o);
                if (StaticClass.debug) {
                    System.out.println("HashtoList { " + o.getKey() + ", " + o.getVal() + " }");
                }
            }
        } else {
            if (StaticClass.debug) {
                System.out.println("HashtoList { le hash est vide ! }");
            }
        }
        return list;
    }

    //=====removing dupliques entries=====
    public static List<Integer> removeDuplicates(List<Integer> inputList) {
        Hashtable uniqueStore = new Hashtable<Integer, Integer>();
        List<Integer> finalList = new LinkedList<Integer>();
        for (Integer currValue : inputList) {
            if (!uniqueStore.containsKey(currValue)) {
                uniqueStore.put(currValue, 0);
                finalList.add(currValue);
            }
        }
        return finalList;
    }

    /**
     old formula: max((P1+2*[F])/3),[F]) with P1=10.5;F=11.0
     new formula: max((10.5+2*11.0)/3),11.0)
     */
    public static String replaceFormulaVar(String formule, Hashtable hash) {
```

```

boolean first = true;
StringTokenizer st = new StringTokenizer(formule);
//detection des variables
while (st.hasMoreTokens()) {

    String str = st.nextToken("[");
    System.out.println(str);
    System.out.println("<br>");
    String[] substr = str.split("[");
    //out.println(st1[0]); //la variable a detecter
    // out.println("<br>");
    if (hash.containsKey(substr[0])) {
        String varI = "[" + substr[0] + "]";
        System.out.println("varI:" + varI + "<br>");//varI:[F]
        System.out.println("hash.get(substr[0]).toString():" + hash.get(substr[0]).toString() + "<br>");
        // formule.replace(varI, hash.get(substr[0]).toString());
        //=====test=====

        formule = replaceWord(formule, varI, hash.get(substr[0]).toString());
        //resultat final
        System.out.println(formule);

    }

} //fin du while
return formule;
}

public static String replaceWord(String original, String find, String replacement) {
    int i = original.indexOf(find);
    if (i < 0) {
        return original; // return original if 'find' is not in it.
    }

    String partBefore = original.substring(0, i);
    String partAfter = original.substring(i + find.length());

    return partBefore + replacement + partAfter;
}

public static Date StringToDate(String str_date) {
    DateFormat formatter;
    Date date = null;
    // String str_date="11-June-07";
    try {
        formatter = new SimpleDateFormat("dd-MMM-yy");
        date = (Date) formatter.parse(str_date);
        //System.out.println("Today is " + date);
    } catch (ParseException e) {
        System.out.println("Exception :" + e);
    }
    return date;
}
}

```

Annexe 3 Filter – Application « HoraireIsaeTC »

```
/**
 *
 * @author salim
 */
public class NewSimpleFilter implements Filter {

    private static final boolean debug = true;

    // The filter configuration object we are associated with. If
    // this value is null, this filter instance is not currently
    // configured.
    private FilterConfig filterConfig = null;

    public NewSimpleFilter() {
    }

    private void doBeforeProcessing(ServletRequest request, ServletResponse response)
        throws IOException, ServletException {
        if (debug) {
            log("NewSimpleFilter:DoBeforeProcessing");
        }

        // Write code here to process the request and/or response before
        // the rest of the filter chain is invoked.

        // For example, a logging filter might log items on the request object,
        // such as the parameters.
        /*
        for (Enumeration en = request.getParameterNames(); en.hasMoreElements(); ) {
            String name = (String)en.nextElement();
            String values[] = request.getParameterValues(name);
            int n = values.length;
            StringBuffer buf = new StringBuffer();
            buf.append(name);
            buf.append("=");
            for(int i=0; i < n; i++) {
                buf.append(values[i]);
                if (i < n-1)
                    buf.append(",");
            }
            log(buf.toString());
        }
        */

    private void doAfterProcessing(ServletRequest request, ServletResponse response)
        throws IOException, ServletException {
        if (debug) {
            log("NewSimpleFilter:DoAfterProcessing");
        }
    }
}
```

```

    }

    // Write code here to process the request and/or response after
    // the rest of the filter chain is invoked.

    // For example, a logging filter might log the attributes on the
    // request object after the request has been processed.
    /*
    for (Enumeration en = request.getAttributeNames(); en.hasMoreElements(); ) {
        String name = (String)en.nextElement();
        Object value = request.getAttribute(name);
        log("attribute: " + name + "=" + value.toString());
    }
    */

    // For example, a filter might append something to the response.
    /*
    PrintWriter respOut = new PrintWriter(response.getWriter());
    respOut.println("<P><B>This has been appended by an intrusive filter.</B>");
    */
}

/**
 *
 * @param request The servlet request we are processing
 * @param response The servlet response we are creating
 * @param chain The filter chain we are processing
 *
 * @exception IOException if an input/output error occurs
 * @exception ServletException if a servlet error occurs
 */
public void doFilter(ServletRequest request, ServletResponse response,
    FilterChain chain)
    throws IOException, ServletException {

    if (debug) {
        log("NewSimpleFilter:doFilter()");
    }

    doBeforeProcessing(request, response);
    //filterConfig.getServletContext().getRequestDispatcher("login.jsp").forward(request, response);
    Throwable problem = null;
    try {
        //System.out.println("new filter -----le debut des appels");
        //response.getWriter().println("Pas disponible ... revenir plus tard");

        HttpServletRequest httpRequest = (HttpServletRequest) request;
        HttpServletResponse httpResponse = (HttpServletResponse) response;

        //do this only when request is not for login.jsp
        if (httpRequest.getRequestURI().indexOf("login.jsp") == -1) {
            HttpSession session = httpRequest.getSession();
            AnCeBean param = (AnCeBean) session.getAttribute("param");
            //getting bean from isae version 0 project as a

```

```

ParamLogin b = (ParamLogin) session.getAttribute("b");

InitialQuerys iq = (InitialQuerys) getFilterConfig().getServletContext().getAttribute("iq");
if (param != null) {
    if (param.getNoChef() == 0 && param.getNoEmp() == 0) {
        httpresponse.sendRedirect("login.jsp");
    }
    //le bean du isae version zero,si null rediriger au login
} else if (b != null) {
    //si le chef n'est pas loggee-----
    if (b.getNo() == 0) {
        if(StaticClass.debug)System.out.println("++++filter++++ bean: b ,nochef="+b.getNo()+" --
redirect to login");
        // filterConfig.getServletContext().getRequestDispatcher("/login.jsp").forward(request, response);
        httpresponse.sendRedirect("login.jsp");

    } else {
        if(StaticClass.debug)System.out.println("++++filter++++ bean: b ,nochef="+b.getNo()+" --redirect
to login");
        param.setNoChef(b.getNo());

        String pn = iq.avoirChef(param.getNoChef());
        param.setPrenomnomchef(pn);
    }
    //si le chef est logger
} //fin si b!=null
else {
    filterConfig.getServletContext().getRequestDispatcher("/login.jsp").forward(request, response);
    // httpresponse.sendRedirect("login.jsp");
}

} //fin de if uri != login
if(StaticClass.debug)System.out.println("++++filter++++2/2 procede chaine");
//procede if user is authenticated

chain.doFilter(request, response);
}

```

Liste des figures

Figure 1: Organigramme de l'ISAE-Cnam Liban	16
Figure 2: Duplication des bases de données aux centres.....	23
Figure 3: Architecture classique	37
Figure 4: Architecture Java.....	37
Figure 5: Presentation JSP	41
Figure 6: Etapes d'une demande de page JSP	43
Figure 7: Le système d'information soumis à deux types de contraintes.....	52
Figure 8: Le processus de développement en Y	53
Figure 9: Paradigme composant à différents niveaux d'abstraction d'un système	58
Figure 10: Exemple d'un modèle composant	60
Figure 11: Exemple d'un modèle composant implémenté à l'institut.....	60
Figure 12: Diagramme de contexte (boîte noire).....	66
Figure 13: Diagramme de contexte (boîte blanche)	67
Figure 14: Diagramme de flux de l'inscription	69
Figure 15: Cas d'utilisation de l'inscription	70
Figure 16: Modélisation des épreuves	74
Figure 17: Architecture JSP/MVC.....	77
Figure 18: Architecture JSP sans séparation des couches	78
Figure 19: Positionnement d'AJAX	80
Figure 20: L'application « Resultat » intégrant AJAX	80
Figure 21: Serveur Tomcat : fenêtre d'administration des applications Web.....	81
Figure 22: Schéma de composants d'inscription.....	82
Figure 23: Projets indépendants avec répétition de codes.....	83
Figure 24: Modèle composant avec échanges d'informations	84
Figure 25: Modèle composants avec deux sources d'informations.....	85
Figure 26: Modèle composants avec une seule source d'informations	85
Figure 27: Sécurité par les filtres.....	87
Figure 28: Première version du SI ISAE-Cnam Liban.....	89
Figure 29: Deuxième version du SI ISAE Cnam Liban	90
Figure 30: Préinscription	91
Figure 31: Automate d'inscription.....	92
Figure 32: Gestion Horaire	94
Figure 33: Stage.....	95
Figure 34: Interface chef (liste des accords).....	95
Figure 35 : Architecture « Jasper Report »	98
Figure 36: Fichier JRXML	98
Figure 37: « Servlet » (Jasper->Pdf).....	100
Figure 38: Rapport.pdf	100

Liste des tableaux

Table 1: Listes des logiciels libres et leurs équivalents propriétaires	27
Table 2 : Comparaison entre JSP et PHP	44
Table 3 : Comparatif détaillé entre MySQL, PostgreSQL, Ingres, Firebird et Zodb	47
Table 4: Comparaison entre MySQL, PostgreSQL, Ingres, Firebird et Zodb	48
Table 5: Type droit.....	75

Participation au développement du Système d'Information de l'ISAE-Cnam Liban, stratégies de développements et architecture.

Mémoire d'ingénieur C.N.A.M., centre régional associé de Beyrouth 2010

RESUME

Ce projet pose la problématique de l'ancien Système d'Information de l'ISAE-Cnam Liban, en vue de trouver les meilleures solutions, et il consiste à développer le nouveau SI.

Les techniques et les technologies utilisées:

- La programmation orientée composant (POC)
- L'architecture N-tiers
- L'Open Source

Mots clés :

Institut, ECTS, crédit, inscription, horaire, open source, JSP, MySQL, Programmation Orientée Composant, langage de modélisation unifié, 2TUP, Model Vue Contrôleur.

SUMMARY

This project poses the problematic of the old ISAE-Cnam Liban Information System in order to find the best solutions and to develop the new IS.

Techniques and technologies used:

- Component-Oriented-Programming
- N-tiers architecture
- Open Source

Keywords:

Institut, ECTS, credit, inscription, schedule, open source, JSP, MySQL, Component-Oriented-Programming, Unified Modeling Language, 2TUP, Model View Controller.